2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

OctHilNet: Hilbert-Guided Hierarchical Geometry Codec for Octree-Structured LiDAR Point Clouds

Mingjian Feng, Mingyue Cui†, Yuyang Zhong, Chunjie Shu, Han Liu, Daosong Hu, and Kai Huang

***Abstract*— High-quality LiDAR point cloud (LPC) compression is essential for the storage and transmission of 3D data. The octree-structured entropy codec has emerged as the predominant method; however, previous methods do not fully utilize spatial contextual information, due to the loss of local features caused by uneven scanning density. To address this problem, we propose OctHilNet, a novel Hilbert-guided hierarchical framework for LPC compression that introduces the polarized octree for efficient node organization and the serialize-driven entropy model to strengthen the continuity of node contexts. Specifically, to counteract the inherent density imbalance, OctHilNet first transforms points into polar coordinates and applies a non-linear rebalancing to the radial distance. Then, we introduce the Hilbert space-filling curve to mitigate the impact of the decoupling between sequential adjacency and geometric proximity in octree node sequences. Finally, to better capture fine-grained spatial correlations, we propose LocAtten and NeighbConv modules in a hierarchical Transformer, which jointly strengthen local dependencies overlooked by standard self-attention. Compared to the previous state-of-the-art works, our method achieves 45.1%-50.1% and 51.9%-53.9% BD-Rate gains on the LPC benchmark SemanticKITTI and MPEG-specified Ford datasets, respectively. In particular, our OctHilNet allows for extension to downstream tasks (i.e., vehicle detection and semantic segmentation), further demonstrating the practicality of the method.**

I. Introduction

LiDAR serves as an active remote sensing device [1], capable of accurately capturing the three-dimensional geometry of surrounding environments. Due to its high accuracy and resolution, it has been widely used in various fields of intelligent robots, such as detection, segmentation, planning, etc. However, at the same time, LiDAR also brings a huge amount of data and the resulting high costs. For example, a single Velodyne LiDAR of the HDL64 model generates over 100,000 points per sweep, resulting in approximately 2.88 million points per second [2]. Different from its well-studied image and video counterparts, storing and transmitting such a large amount of LPC data remains a serious challenge due to the disorder and sparsity of the point cloud.

Fortunately, several schemes for LPC geometry compression, such as voxel-based methods and octree-based methods, have been explored and applied in prior research works [3], [4]. The MPEG point cloud geometry compression standard (GPCC) [5] adopts a hand-crafted context-adaptive arithmetic encoder for efficient bit allocation. Quach *et al.* [6] propose a voxel-based geometry compression method for point clouds, which uses a convolutional auto-encoder to process voxelized features, and uniform quantization to decode grid occupancies. Kaya et al. [7] further introduce voxel-based convolutional transforms for lossy point cloud geometry compression to refine bounding volumes in voxelized point clouds. Although voxel-based methods can utilize local geometric patterns (*e.g.*, planes, surfaces), they are strictly coupled with resolution size and receptive fields are limited by the computational cost, which allows them to only extract features from voxels within a narrow range. Recently, OctSqueeze [8] proposes the first octree-based deep learning entropy model by leveraging context formation about conditions on ancestor nodes to discover the dependency, making it the predominant method [9], [10], [11]. However, previous studies focus more on large-scale attention-based context prediction, overlooking the loss of local features caused by uneven scanning density, which is significant to exploit the geometry redundancy. Moreover, these methods typically serialize octree nodes in Z-order curve [12], which inherently decouples sequential adjacency from geometric proximity, making it difficult to capture fine-grained local nodes' features. In general, prior voxel-based and octree-based methods do not fully use the spatial context information of LPCs.

In this paper, we propose OctHilNet, a Hilbert-guided hierarchical geometry codec for octree-structured LiDAR point clouds. OctHilNet combines the polarized octree for

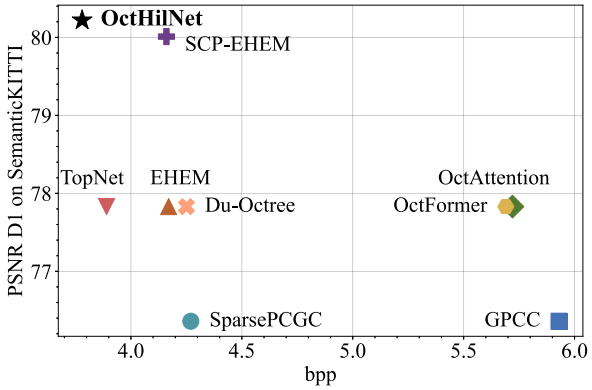

Fig. 1: Bits per point (bpp) [9] and PSNR D1 [13] of our OctHilNet and other baselines on the SemanticKITTI dataset [14].

*This work was supported in part by the Guangdong Hong Kong Macao Applied Mathematics Center Project, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2025A1515011485, and in part by the Guangxi Key R & D Program under Grant No.GuikeAB24010324.

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

†Corresponding author.

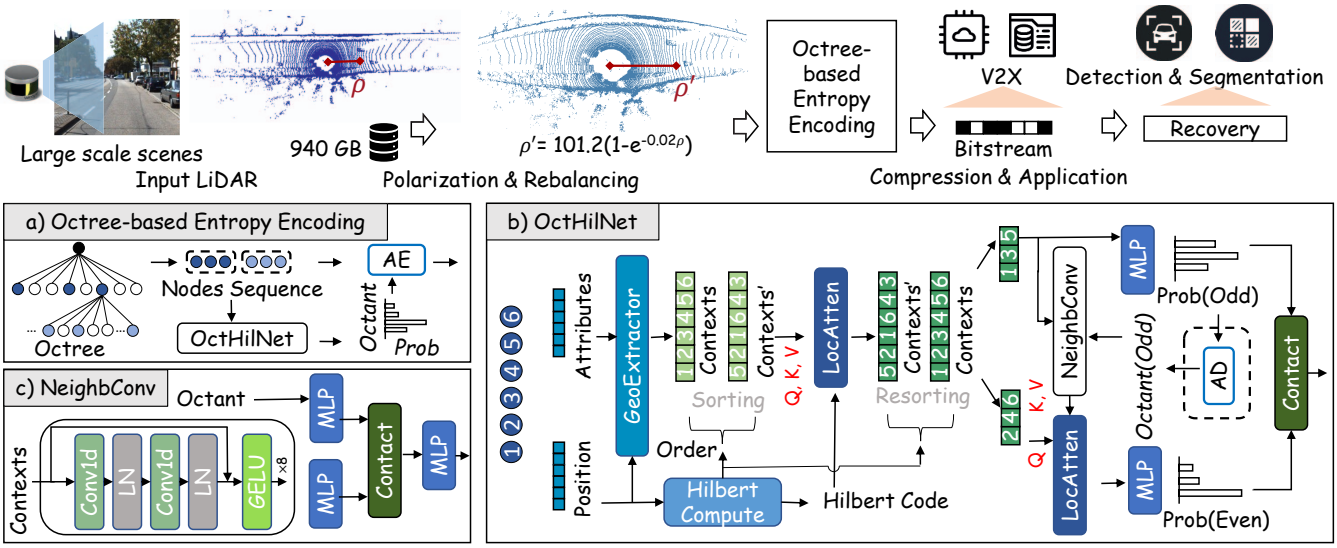

Fig. 2: Framework of the proposed method. (a) Process of the octree-based entropy encoding. (b) Overview of OctHilNet. (c) Structure of NeighbConv module.

efficient node organization and the serialize-driven entropy model for enhanced continuity of node contexts, effectively exploiting spatial context information in LPCs. Specifically, we introduce a polar coordinate transformation with non-linear radial rebalancing to address the inherent density imbalance of point distributions, achieving higher quality with fewer nodes in octree construction. To address the spatial-to-sequential mismatch of octree nodes' sequences, we further adopt a Hilbert space-filling curve for re-organization, which ensures that the adjacent nodes are geometrically proximal and improve the model's ability to capture the neighboring contexts. Furthermore, to extract fine-grained contexts from the spatially adjacent nodes, we propose LocAtten module to exploit the restored locality with an attention mechanism and depth-wise convolutions. By incorporating sorted Hilbert codes into the Transformer's positional bias, the LocAtten module explicitly captures the geometric information. Moreover, we also propose the NeighbConv module with convolutions and residual connections to strengthen interactions between neighboring elements.

We compare our OctHilNet with state-of-the-art methods such as GPCC [15], SparsePCGC [16], OctAttention [9], OctFormer [10], DuOctrtee [2], EHEM [17], SCP-EHEM [11], and TopNet [1] on the LPC benchmark SemanticKITTI [14] and MPEG-specified Ford [18] datasets and downstream tasks (*i.e.*, vehicle detection and semantic segmentation). The experiments show that our method outperforms these methods, as illustrated in Fig. 1. Overall, our main contributions are summarized as follows:

- We propose OctHilNet, a novel Hilbert-guided hierarchical attention mechanism that introduces the polarized octree for efficient node organization and the serialize-driven entropy model to strengthen the continuity of node contexts.
- We innovatively apply a non-linear rebalancing to the radial distance on the polar coordinate system to counteract the inherent density imbalance, which achieves higher quality with fewer nodes in octree construction.
- To mitigate the impact of the decoupling between sequential adjacency and geometric proximity, we introduce a Hilbert space-filling curve to organize the octree nodes.
- To capture fine-grained spatial correlations, we propose LocAtten and NeighbConv modules in a hierarchical Transformer by incorporating Hilbert codes and convolutions.

II. Related Work

A. Voxel-based Methods

Voxel-based methods [19], [20] usually discretize point clouds into voxel grids and apply 3D convolution to predict the occupancy of each voxel. Wang *et al.* [21] propose an end-to-end learned point cloud geometry compression system that voxelizes point clouds into non-overlapping 3D cubes and leverages stacked deep neural networks based on Variational AutoEncoder (VAE) to compress the point clouds. Voxel-FPN [22] further proposes a one-stage trainable deep architecture for multi-scale voxel partitioning, which receives multiple scales of voxel grids and decodes them via a top-down pyramid network. Voxel representations can preserve the geometric structures of point clouds; however, they are sensitive to density variations and incur high computational costs when handling sparse point clouds.

B. Octree-based Methods

Unlike voxel-based coding, octree-based methods [3], [4] leverage a larger receptive field through efficient node organization and tree-structured long context modeling, which improves compression performance. GPCC [5] proposes a hand-crafted context-adaptive arithmetic encoder for octree bit allocation, which predicts the encoding node based on the previously coded information. OctSqueeze [8] introduces the first octree-based deep learning entropy model, which captures dependencies by using convolution blocks for ancestor nodes. SparsePCGC [16] promotes it with

sparse convolutional networks on multiscale sparse tensors, leveraging correlations through cross-scale context modeling. OctAttention [9] then employs a conditional entropy model to capture neighboring node dependencies and leverage a large receptive field with masking to enable the encoding of multiple nodes. Based on OctAttention, OctFormer [10] refines the frequent multi-head self-attention operation by sharing results across constructed non-overlapped windows.

Recently, EHEM [17] proposes adopting grouped attention to extract features from the large-scale context of ancestor and sibling nodes to reduce the compression redundancy, and serial coding to accelerate the decoding process. Building on this, SCP-EHEM [11] integrates a spherical octree and employs an extent-level context module to mitigate reconstruction errors in distant voxels. Without performing coordinate transformation, DuOctree [2] proposes a dual-octree structure for LPC representation and incorporates a cross-attention entropy model to capture hierarchical spatial dependencies. TopNet [1] designs a local-enhanced context encoding for enhancing translation-invariance, and adaptive sliding window attention for discovering multi-scale dependencies. However, the above works focus more on large-scale attention-based context modeling but often overlook the uneven density, which leads to the loss of local features. As a result, the spatial context is not fully exploited, which limits compression efficiency.

III. Method

A. Framework

As shown in Fig. 2, we propose a Hilbert-guided hierarchical attention mechanism called OctHilNet, which is an octree-structured LPC entropy compression framework. Specifically, we first transform the point cloud into polar coordinates and apply a nonlinear radial rebalancing operation to alleviate the severe density imbalance inherent in LPCs, which helps preserve local details. The rebalanced data is then organized into an octree representation. Considering that the neighboring nodes of the octrees in 3D space are placed far apart in the standard node sequence, we introduce the Hilbert space-filling curve to mitigate the impact of the decoupling between sequential adjacency and geometric proximity in octree node sequences. Node positions are mapped to Hilbert codes via the Hilbert compute module, with sorted codes defining the Hilbert order. This operation organizes spatially adjacent nodes in the sequence, enhancing the spatial coherence of the context for the entropy model. Besides, we design a hierarchical Transformer, integrated with LocAtten and NeighbConv modules, to better capture the fine-grained local dependencies within the node sequences. Finally, the distribution of each node's occupancy is predicted, and an arithmetic encoder (AE) is used for lossless compression of octree occupancy sequences.

B. Polarization and Rebalancing

LPCs inherently exhibit a non-uniform density, characterized by high point concentrations near the sensor and sparsity at larger distances. This non-uniformity impairs hierarchical partitioning methods such as the octree. To address this, we first transform the point cloud into polar coordinates and subsequently apply a non-linear function to rebalance the radial distance ρ, yielding a more uniform point distribution that facilitates octree construction. Specifically, we employ a normalized inverse exponential function on the radial component of the polarized coordinates, which guarantees a consistent maximum radial distance to preserve the original scale. The transformation is defined as:

$$\rho' = (1 - e^{-m\rho}) \cdot \frac{\rho_{\max}}{1 - e^{-m\rho_{\max}}} \tag{1}$$

where ρ_{max} is the maximum radial distance in the point cloud and m is a hyperparameter controlling the transformation. Appropriate choices of m empirically lead to a more uniform point distribution.

After rebalancing the LPC data, we construct the octree following [11]. The construction process divides the space into 8 cubes based on 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 octant to represent occupancy status of its children. The octree nodes are then organized into sequences via a breadth-first traversal, which spatially corresponds to the Z-order curve [23].

C. Context Entropy Model

We use $X = [x_1, x_2, ..., x_i, ...x_N]$ to represent a sequence of octants for octree nodes, where x_i is the octant of the i_{th} octree node. Each octant $(1 - 255)$ represents the decimal value of its 8-bit binary code. Furthermore, we can factorize the parametric probability distribution $Q(X)$ into a product of predicted probability distributions of each octant x_i:

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

where $q_i(x_i|\mathbf{f_i}; w)$ is the estimated distribution of octant x_i and w is the weight of the entropy model. The context vector $\mathbf{f}_i$ is constructed by concatenating the feature vectors of its ancestors and its already decoded siblings in the sequence:

$$\mathbf{f}_i = \text{concat}([\mathbf{a}_1, .., \mathbf{a}_k, ..\mathbf{a}_K], [\mathbf{v}_{i-N+1}, .., \mathbf{v}_j, ..\mathbf{v}_{i-1}]) \tag{3}$$

where $\mathbf{a}_k$ represents the feature vector of the k-th ancestral node, and $\mathbf{v}_j$ is the feature vector of the j-th previously decoded sibling node at the same level, as described in [17]. Each node's feature vector $\mathbf{v}_j$ includes its position and attributes, such as xyz coordinates, index (0-7), cube extents, depth (1-14), and the node's octant (1-255).

OctHilNet minimizes the cross-entropy $H(\hat{Q}, Q) = \mathbb{E}_{X\sim\hat{Q}}[-\log_2 Q(X)]$, where $\hat{Q}$ is the true distribution of octree sequences and Q is the predicted distribution. By reducing $H(\hat{Q}, Q)$, the expected bitrate approaches the Shannon entropy, i.e., the theoretical lower bound of compression [24].

D. Hilbert-based Attention Entropy Model

1) Hilbert Sorting: Due to the sparsity of LPCs, it is difficult for typical octrees to capture local geometric relationships, because their node sequences are arranged in

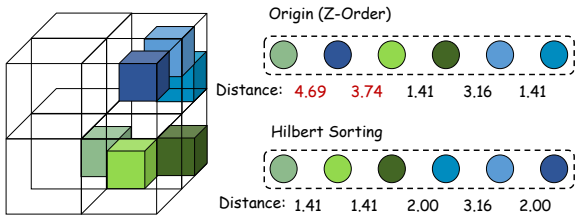

Fig. 3: Hilbert sorting on octree node sequences.

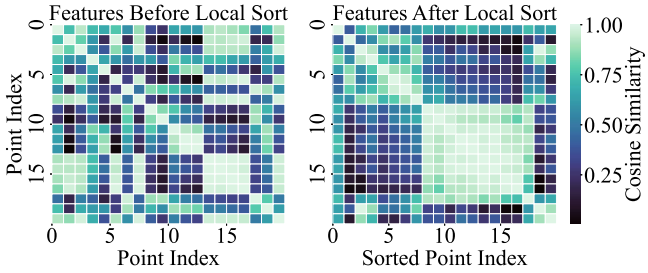

Fig. 4: Comparison of context after Hilbert sorting.

Morton code-based Z-order curves [25]. As shown in Fig. 3, the geometric distance between neighbor nodes from the original octree sequence is larger than that in the sorted sequence. Because the adjacent node positions and the geometric information they represent are not fully correlated, it is difficult to extract local geometric features from the sequence. Hilbert order [12], [26] adheres to the rotationally consistent U-shaped connection rule, which has superior locality-preserving properties compared with the Z-order curve. In OctHilNet, we employ a DGCNN-based [27] geometry extractor to process the input attributes and positional encoding to generate rich contextual features. To efficiently serialize the contextual features, we calculated the Hilbert code for the position and sorted them based on Hilbert order. After sorting, the similarity of local context increases, which enhances the local features, as shown in Fig. 4. For consistent encoding and decoding, the sorted sequence needs to be restored to its original order before entropy coding.

2) LocAtten Module: To effectively capture local geometric dependencies within Hilbert-sorted feature sequences, we propose the LocAtten module, which is designed to enhance local spatial features. Window-based operations are employed to handle both local and long-range dependencies efficiently, as shown in Fig. 5(a).

In the module, we first employ a contextual splitting operation to divide serialized sequences into local contexts. Vanilla self-attention treats inputs as fully connected, ignoring the local structures in spatial data. To effectively leverage the spatial features, we propose a Hilbert-enhanced local attention mechanism, illustrated in Fig. 5(b) and Fig. 5(c). It incorporates both local context and a structural prior into the attention computation, and includes a LocalConv module to further refine local feature representations. To better capture local feature patterns, we then augment the query, key, and value projections by fusing the standard linear transformations with parallel depth-wise 1D convolutions.

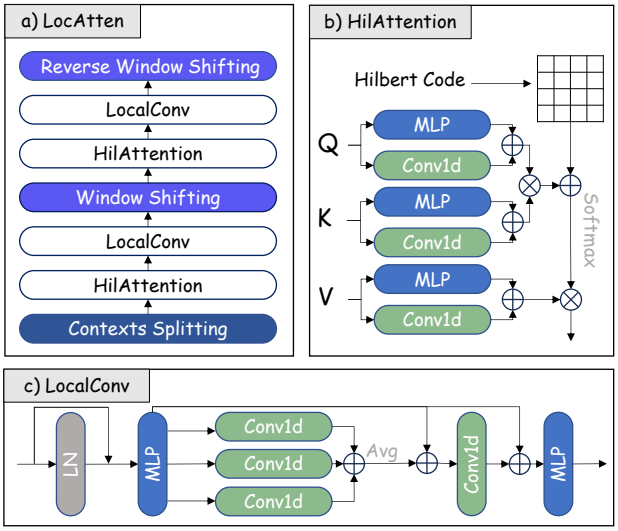

Fig. 5: The structure of the LocAtten module. (a) LocAtten module applies context splitting, shifted window operations, HilAttention, and LocalConv modules. (b) HilAttention module leverages Hilbert code to compute attention map. (c) LocalConv module enhances feature extraction through 1D convolutions and residual fusion.

The augmented query $\mathbf{Q}'$ is computed as:

$$\mathbf{Q}' = \mathbf{f}\mathbf{W}_{\mathbf{Q}} + \text{Conv1D}(\mathbf{f}) \tag{4}$$

where $\mathbf{f}$ represents the input feature sequence and $\mathbf{W}_{\mathbf{Q}}$ is the weight matrix for the query projection. The $\mathbf{K}'$ and $\mathbf{V}'$ representations are computed analogously. We further introduce Hilbert Proximity Prior $\mathbf{P}$ to leverage the locality-preserving ordering, which is dynamically computed from the pairwise distances of smoothed Hilbert codes c'. This prior encourages higher attention scores between spatially adjacent tokens. The prior is calculated as:

$$\mathbf{P}_{ij} = (1 - \text{NormDist}(\mathbf{c'}_i, \mathbf{c'}_j))(1 - p) + p \tag{5}$$

where $\text{NormDist}(\cdot, \cdot)$ is the L2 distance between the smoothed Hilbert codes of tokens i and j, and p is a hyperparameter for establishing the minimum prior. Finally, this prior modulates the relative position bias $\mathbf{B}$. The complete attention output is then computed by integrating these components with the augmented value representation $\mathbf{V}'$:

$$\text{Attention} = \text{Softmax}\left(\frac{\mathbf{Q}'(\mathbf{K}')^{\top}}{\sqrt{d_k}} + \mathbf{B} \odot \mathbf{P}\right)\mathbf{V}' \tag{6}$$

where d_k is the dimension of the key vectors, $\mathbf{B}$ is the relative position bias, and $\odot$ denotes the element-wise product. Moreover, inspired by vision transformers for 2D images [28], we introduce window shifting and reverse shifting between attention layers to capture cross-window dependencies, which enlarges the receptive field of local attention.

3) NeighbConv module: We design NeighbConv to extract neighborhood information from sibling groups effectively. As shown in Fig. 2(b) and Fig. 2(c), the module incorporates 1D convolutions and residual connections for capturing neighbor dependencies, thereby enhancing local

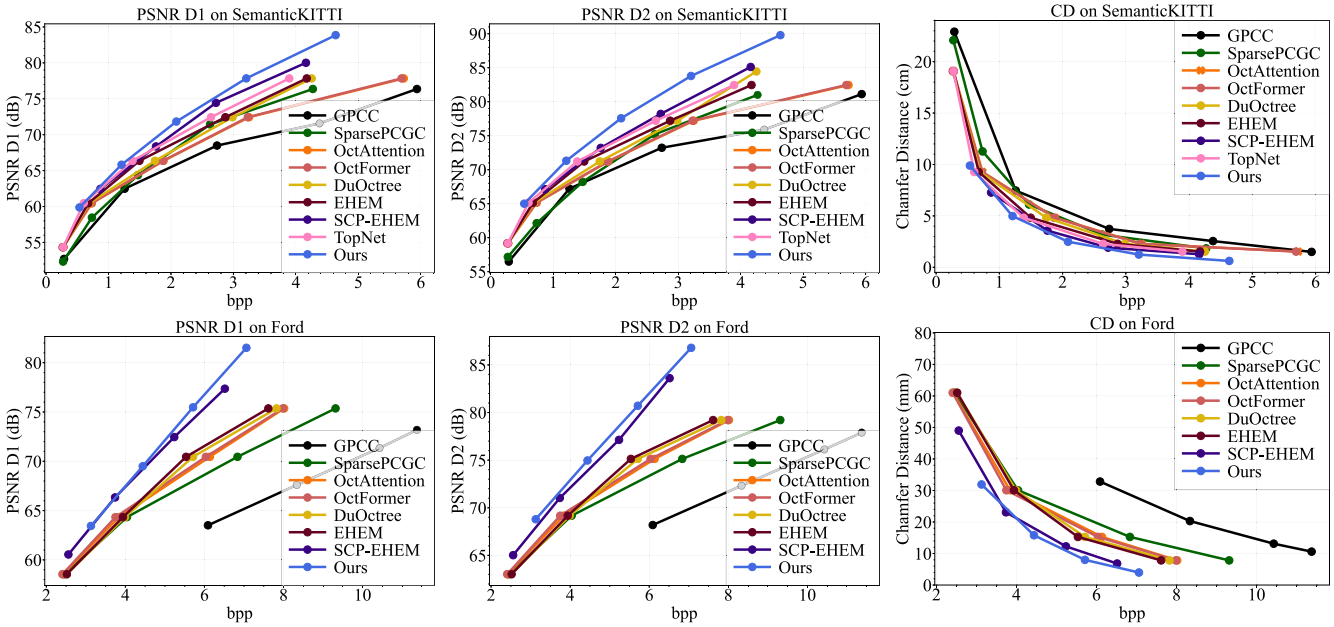

Fig. 6: Quality results of different compression methods on the SemanticKITTI and Ford datasets at different bitrates.

TABLE I: BD-Rate gains over baseline GPCC measured using PSNR (D1, D2), and CD for ours and other methods.

Method	Reference	SemanticKITTI			Ford		
		PSNR D1	PSNR D2	CD	PSNR D1	PSNR D2	CD
SparsePCGC [16]†	TPAMI'22	-15.71%	-10.84%	-6.59%	-34.05%	-34.08%	-33.27%
OctAttention [9]‡	AAAI'22	-22.01%	-24.18%	-24.61%	-39.55%	-39.63%	-38.53%
OctFormer [10]†	AAAI'23	-22.92%	-25.05%	-25.96%	-40.75%	-40.74%	-39.77%
EHEM [17]‡	CVPR'23	-31.58%	-33.56%	-29.69%	-42.68%	-42.53%	-41.32%
SCP-EHEM [11]‡	AAAI'24	-40.63%	-40.30%	-37.76%	-51.49%	-51.35%	-51.22%
DuOctree [2]†	ICRA'24	-27.34%	-29.62%	-28.32%	-41.31%	-41.16%	-39.99%
TopNet [1]‡	CVPR'25	-36.96%	-38.81%	-35.31%	-	-	-
Ours‡	-	**-45.06%**	**-50.12%**	**-42.31%**	**-51.92%**	**-53.86%**	**-52.22%**

† parent-introduced that only depends on the parent context, ‡ sibling-introduced that depends on both parent and sibling contexts.

feature within partitioned sibling groups. The octant sequence $\mathbf{x}$ is partitioned into odd-positioned $\mathbf{x}_{i1}$ and even-positioned $\mathbf{x}_{i2}$ [17], which enables parallel decoding by using sibling node octant as the feature. The NeighbConv module receives attention contexts and octants $\mathbf{x}_{i1}$ of odd-positioned neighbors as input. It generates key–value vectors for LocAtten module in predicting the octants $\mathbf{x}_{i2}$ of even-positioned nodes. $\mathbf{x}_{i1}$ is directly obtained during encoding and recovered via the arithmetic decoder (AD) during decoding, relying solely on odd-positioned features. Our NeighbConv performs convolutions within sibling groups to extract neighborhood information, which reduces the disruption of local geometric dependencies and fully exploits context.

IV. Experiments

A. Dataset

1) SemanticKITTI: SemanticKITTI [14] is a widely used large-scale outdoor dataset collected by a 64-laser LiDAR. It comprises 22 sequences with a total of 43,504 scans, each captured at 10 Hz and containing approximately 128,000 points. Following the standard split [1], we use sequences 00–10 for training, and sequences 11–21 for testing.

2) Ford: Ford [18] is another LiDAR dataset commonly used in MPEG point cloud compression [5] that contains more than 1,500 scans per sequence. We follow the partitioning guidelines recommended by the MPEG standardization [15], designating Sequence 01 for model training and setting Sequences 02 and 03 for performance evaluation.

B. Experimental Details

1) Baselines: We evaluate the proposed method against state-of-the-art methods, including GPCC [15] (TMC13 v27.0), the parent-introduced methods SparsePCGC [16], OctFormer [10], and DuOctree [2], as well as the sibling-introduced methods OctAttention [9], EHEM [17], SCP-EHEM [11], and TopNet [1]. These methods are designed for a specific category of point clouds. To ensure fair comparison, we adopt the same training and testing settings as these baselines.

2) Metrics: We evaluate reconstruction quality using point-to-point PSNR (PSNR D1) and point-to-plane PSNR (PSNR D2) [13], and measure compression efficiency with bits per point (bpp) [9]. In addition, we report Chamfer Distance (CD) [29]. BD-Rate [30] is used to evaluate rate–distortion performance, with GPCC as the anchor. Unless otherwise stated, all distortion curves and bitrates are

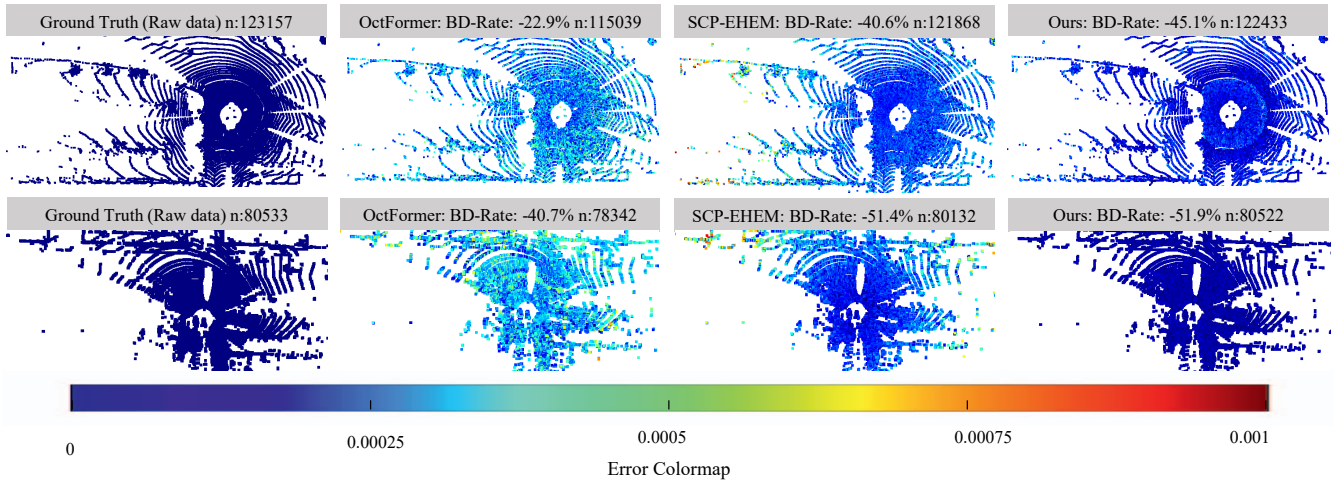

Fig. 7: Visualization of our OctHilNet and other baselines with different encoded point numbers n on the SemanticKITTI (upper) and Ford (bottom) datasets.

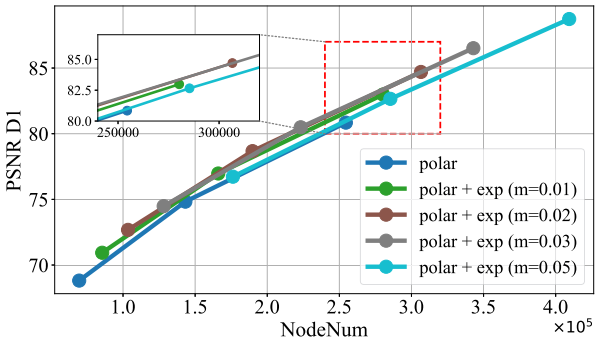

Fig. 8: Node number distortion curve with different exponential functions on the SemanticKITTI dataset.

TABLE II: Ablation study of OctHilNet on the SemanticKITTI. Default settings are marked in gray.

HS	LA	ST	NC	MLP	bpp ↓				BD-Rate
	✓		✓		1.41	2.32	3.55	4.92	-34.59%
✓		✓	✓		1.35	2.27	3.50	4.89	-36.99%
✓	✓			✓	1.34	2.23	3.44	4.78	-37.89%
✓	✓		✓		**1.20**	**2.08**	**3.20**	**4.63**	**-45.06%**

HS: Hilbert Sorting, LA: LocAtten, ST: SwinTransformer [28], NC: NeighbConv, MLP: Multilayer Perceptron.

averaged over sequences.

3) Implementation Details: For fair evaluation, we follow [17], [11] for the quantization step on the SemanticKITTI ($400/2^{D-1}$) and Ford (2^{18-D}), with an octree sequence size of 8,192 and maximum depth 14 for both datasets. We train OctHilNet for 16 epochs using AdamW (learning rate=1e-4, weight decay=1e-2) with step decay γ=0.3 every 6 epochs. Experiments are implemented in PyTorch on an Intel Xeon Gold 6234 CPU and a NVIDIA RTX 4090 (48GB). We set m to 2e-2 and 1e-5, respectively on the SemanticKITTI and Ford dataset in the inverse exponential function for the polarized coordinates. Our hierarchical attention model consists of 5 blocks with 4, 4, 4, 4, and 2 layers. The hyperparameter p in the Hilbert proximity prior is set to 0.5. The loss function is defined as the cross-entropy between the predicted distribution and the real distribution of each octree node. The downstream application models are trained on raw point clouds and tested on decoded outputs.

TABLE III: Complexity comparison of different methods on the SemanticKITTI dataset.

Method	Ref.	Mem.	Enc.	Dec.
SparsePCGC [16]	TPAMI'22	0.60GB	0.52s	0.33s
OctAttention [9]	AAAI'22	0.37GB	0.32s	321s
OctFormer [10]	AAAI'23	2.45GB	0.73s	1.02s
EHEM [17]	CVPR'23	0.85GB	1.48s	1.57s
SCP-EHEM [11]	AAAI'24	0.86GB	2.34s	2.41s
DuOctree [2]	ICRA'24	1.43GB	4.91s	5.02s
TopNet [1]	CVPR'25	0.33GB	2.81s	506s
Ours	-	0.35GB	4.71s	5.03s

C. Main Results

The rate–distortion curves for LiDAR compression are presented in Fig. 6, while Table I reports the BD-Rate gains over GPCC, computed using PSNR D1, PSNR D2, and CD for all evaluated methods. From the figure and table, we can observe that our method outperforms other methods on both the SemanticKITTI and Ford datasets, as expected. This is because we propose the polarized octree for efficient node organization and the serialize-driven model to strengthen the continuity of node contexts. Specifically, compared with GPCC, we achieve gains up to 45.06%, 50.12%, and 42.31% on SemanticKITTI, as well as 51.92%, 53.86%, and 52.22% on Ford. Another observation is that SCP-EHEM performs better than other methods, which may benefit from the spherical coordinates for efficient octree organization, and large-scale window attention to discover long-range dependencies within nodes. Nevertheless, our method still achieves up to 4.4% and 0.4% PSNR D1 gains compared with them. The potential reason is that, on one hand, we employ nonlinear mapping to improve the efficiency of point cloud organization. On the other hand, we introduce Hilbert-based attention, enabling the local enhancement module to better capture spatial correlations within the reordered context. We further notice that when comparing to SCP-EHEM, the BD-Rate gains on SemanticKITTI are higher than Ford, likely because

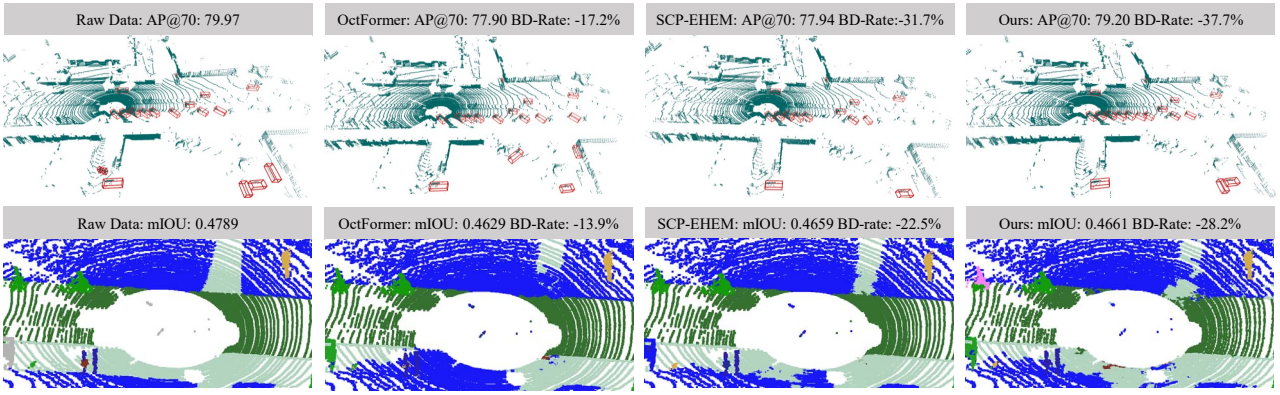

Fig. 9: The visualization of vehicle detection (top) and semantic segmentation (bottom) tasks under different methods.

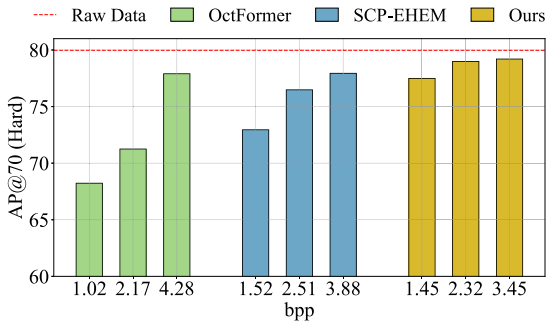

Fig. 10: Quantitative results of vehicle detection at the hard level using different point cloud compression methods.

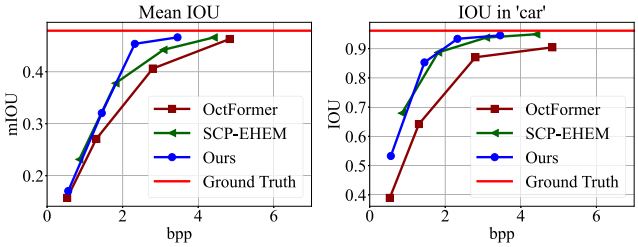

Fig. 11: Quality performance of semantic segmentation within different methods on the SemanticKITTI dataset.

its more concentrated point distribution strengthens spatial contextual dependencies, making node occupancy easier to predict. Besides, we can see that our method produces colors closer to the raw point cloud from Fig. 7, indicating that it reaches superior compression rates while preserving more point details for higher reconstruction quality.

D. Ablation Study

1) Analysis of Rebalancing: As shown in Fig. 8, we compare model performance under different rebalancing functions. We can observe that with exponential rebalancing and m set to 0.02, our method achieves higher reconstruction quality with fewer octree nodes than other values. The improvement is likely because introducing m increases the separability of points with larger radial distances, which helps preserve more details of sparse distant points and enhances reconstruction quality.

2) Analysis of Different Modules: We also compare the model performance with different module configurations, as shown in Table II. First, it can be noted that the combination of our proposed three modules (HS, LA, NC) yields the highest BD-Rate gains, as expected. Moreover, we can see that Hilbert Sorting contributes a gain of 10.46%, while replacing SwinTransformer [28] with LocAtten module and replacing MLP with NeighbConv module bring gains of 8.07% and 7.17%, respectively. These improvements can be attributed to LocAtten module, which better captures serialized node dependencies through local enhancement, and NeighbConv module, which strengthens spatial perception via convolutions, thereby achieving superior compression performance.

3) Analysis of Computation: Furthermore, we perform an ablation study on encoding time, decoding time, and memory usage of our method, as illustrated in Table III. First, we notice that our method uses only 0.35GB of memory, making it one of the most memory-efficient methods. This is mainly due to our LocAtten and NeighbConv modules, which avoid replying to a large receptive field to extract contextual features and save memory overhead. Another observation is that our method requires 8.30s for encoding and 9.15s for decoding, which is higher than other methods. It is mainly attributed to the Hilbert sorting operation, which serializes nodes for the whole sequence, introducing additional computational costs.

E. Performance on Application

1) Vehicle Detection: We conduct 3D vehicle detection experiments using Part-A^2 Net [31]. Following the KITTI benchmark protocol, we report average precision (AP) for 3D vehicle detection with a mean intersection-over-union (mIOU) threshold of 0.7 under the hard level. Results under the hard setting are illustrated in Fig. 9, with quantitative comparisons reported in Fig. 10. We can see that our method achieves the highest AP among all compared methods, while yielding the largest BD-Rate of 37.7%. Furthermore, the detection accuracy remains close to that of raw point clouds, indicating a small impact from compression.

2) Semantic Segmentation: For 3D semantic segmentation, we adopt RandLA-Net [32] as the backbone, and evaluate performance using mIOU and the IoU of the 'Car'

class. Quantitative results are provided in Fig. 11, and representative visualizations are shown in Fig. 9. We observe that OctHilNet achieves segmentation accuracy comparable to that of raw point clouds, particularly at a bitrate of approximately 3.5 bpp, where fine-grained structural details are also well preserved.

In summary, our method outperforms others in both vehicle detection and semantic segmentation tasks, demonstrating its effectiveness in downstream applications.

V. Conclusion

In this paper, we propose a Hilbert-guided hierarchical attention codec, OctHilNet, which innovatively combines radial rebalancing to mitigate density imbalance and a hierarchical Transformer to enhance geometric features in LPCs. By introducing a Hilbert space-filling curve, our method can effectively mitigate the impact of the decoupling between sequential adjacency and geometric proximity in octree node sequences. Besides, we design the LocAtten and NeighbConv modules, which leverage Hilbert code-based positional encoding and convolutions to better capture fine-grained spatial correlations. Experiments demonstrate that OctHilNet significantly outperforms state-of-the-art methods and generalizes robustly to downstream tasks.

References

[1] X. Wang and H. Wang, "Topnet: Transformer-efficient occupancy prediction network for octree-structured point cloud geometry compression," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 27 305–27 314.

[2] M. Cui, M. Feng, J. Long, D. Hu, S. Zhao, and K. Huang, "A du-octree based cross-attention model for lidar geometry compression," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 3796–3802.

[3] S. Biswas, J. Liu, K. Wong, S. Wang, and R. Urtasun, "Muscle: Multi sweep compression of lidar using deep entropy models," in *Proceedings of the 34th International Conference on NeurIPS*. Curran Associates Inc., 2020.

[4] Z. Que, G. Lu, and D. Xu, "Voxelcontext-net: An octree based framework for point cloud compression," in *Proceedings of the IEEE/CVF Conference on CVPR*, 2021, pp. 6042–6051.

[5] D. Graziosi 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*, vol. 9, p. e13, 2020.

[6] M. Quach, G. Valenzise, and F. Dufaux, "Learning convolutional transforms for lossy point cloud geometry compression," in *IEEE International Conference on Image Processing*, 2019, pp. 4320–4324.

[7] E. C. Kaya, S. Schwarz, and I. Tabus, "Refining the bounding volumes for lossless compression of voxelized point clouds geometry," in *IEEE International Conference on Image Processing*, 2021, pp. 3408–3412.

[8] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, "Octsqueeze: Octree-structured entropy model for lidar compression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 1313–1323.

[9] C. Fu, G. Li, R. Song, W. Gao, and S. Liu, "Octattention: Octree-based large-scale contexts model for point cloud compression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[10] M. Cui, J. Long, M. Feng, B. Li, and H. Kai, "Octformer: Efficient octree-based transformer for point cloud compression with local enhancement," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, pp. 470–478, Jun. 2023.

[11] A. Luo, L. Song, K. Nonaka, K. Unno, H. Sun, M. Goto, and J. Katto, "Scp: Spherical-coordinate-based learned point cloud compression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 3954–3962.

[12] V. Pambuccian, "David hilbert. david hilbert's lectures on the foundations of geometry, 1891–1902." *Philosophia Mathematica*, vol. 21, no. 2, pp. 255–277, 2013.

[13] S. Schwarz and M. Preda, "Emerging mpeg standards for point cloud compression," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2019.

[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] Mammou, "Mpeg 3d graphics coding. g-pcc test model v27." in *Output document N18189. ISO/IEC MPEG (JTC 1/SC 29/WG 11)*. 145th MPEG meeting, 2024.

[16] J. Wang, D. Ding, Z. Li, X. Feng, C. Cao, and Z. Ma, "Sparse tensor-based multiscale representation for point cloud geometry compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 7, pp. 9055–9071, 2023.

[17] R. Song, C. Fu, S. Liu, and G. Li, "Efficient hierarchical entropy model for learned point cloud compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 368–14 377.

[18] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *The International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, 2011.

[19] S. Limuti, E. Polo, and S. Milani, "A transform coding strategy for voxelized dynamic point clouds," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 2954–2958.

[20] B. Anand, V. Barsaiyan, M. 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)*, 2019, pp. 1–5.

[21] J. Wang, H. Zhu, H. Liu, and Z. Ma, "Lossy point cloud geometry compression via end-to-end learning," *IEEE Trans on Circuits and Systems for Video Technology*, vol. 31, no. 12, pp. 4909–4923, 2021.

[22] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-fpn: Multi-scale voxel feature aggregation for 3d object detection from lidar point clouds," *Sensors*, vol. 20, no. 3, 2020.

[23] P. Chen and Z. Liu, "Octocache: Caching voxels for accelerating 3d occupancy mapping in autonomous systems," in *Proceedings of the 30th ACM International Conference on ASPLOS*, 2025, pp. 704–718.

[24] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[25] M. Bern, D. Eppstein, and S.-H. Teng, "Parallel construction of quadtrees and quality triangulations," in *Algorithms and Data Structures*, F. Dehne, J.-R. Sack, N. Santoro, and S. Whitesides, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 188–199.

[26] W. Chen, X. Zhu, G. Chen, and B. Yu, "Efficient point cloud analysis using hilbert curve," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 730–747.

[27] 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 Transactions on Graphics (TOG)*, 2019.

[28] 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 *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10 002.

[29] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on CVPR*, 2017, pp. 605–613.

[30] G. Bjøntegaard, "Calculation of average psnr differences between rd-curves," in *ITU-T SG 16/Q6, 13th VCEG Meeting*, April 2001.

[31] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2647–2664, 2020.

[32] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 108–11 117.