Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 20, No. 1, 2026

Structure-Aware and Context-Modeling Point Cloud Compression

Ying He, Chuanjiang Yan, Zhisen Chen

School of Computer Science & Engineering, Chongqing University of Technology, Chongqing 400054, China

Abstract: To address the limited geometric representation capability and the coarse-grained context modeling in learning-based point cloud compression for LiDAR point cloud coding, we propose a structure-aware and context-modeling point cloud compression method (SACM-PCC). On the representation learning side, we design a Structure-Aware Target Embedding module to achieve structural alignment and effective propagation of cross-scale voxel features, thereby enhancing the expression of geometric relationships from local to global. On the probabilistic modeling side, we build a progressive bitwise target occupancy predictor that adopts a conditional autoregressive strategy to decompose each 8-bit occupancy code into four sub-codes and progressively refine the probability estimation from the most significant bits to the least significant bits, improving spatial context utilization and bit-level discrimination accuracy. Experiments on the KITTI and Ford datasets show that, at comparable reconstruction quality, SACM-PCC reduces the bitrate on KITTI by approximately 57%, 21%, and 8.7% relative to Draco, G-PCCv23, and RENO, respectively, and by approximately 54%, 21.7%, and 9% on Ford. These results demonstrate that the proposed method achieves a better rate–distortion trade-off across the full bitrate range while maintaining stable geometric reconstruction performance in complex scenes.

Keywords: LiDAR Point Cloud; Multi-scale Sparse Tensor; Structure-Aware; Progressive Bitwise.

1. Introduction

With the continuous decline in the cost of 3D sensing technology, point clouds have been widely adopted as a versatile data format for efficiently representing three-dimensional objects and scenes across multiple critical domains, including robotics, autonomous driving, augmented reality, and 3D reconstruction. Point cloud data consists of a vast array of discrete points, offering rich geometric structure and semantic information. However, its enormous data volume—particularly pronounced in high-resolution and complex scenes—results in significant bandwidth and resource overhead during storage and transmission. Consequently, point cloud compression methods have remained a focus of ongoing research [1].

In recent years, significant progress has been made in learning-based point cloud geometric compression methods. Compared to traditional rule-based coding schemes, learning-based approaches can simultaneously optimize rate-distortion performance through end-to-end training in both representation learning and probabilistic modeling. This enables higher reconstruction quality at the same coding rate or significantly reduced bit overhead while maintaining comparable reconstruction quality. As a representative work [2], RENO constructs a lightweight neural network coding framework based on multi-scale sparse tensor representations. It avoids the complex hierarchical traversal and inference processes inherent in traditional tree structures. By learning sparse occupancy codes, it enables cross-scale information transfer, thereby reducing inference complexity and latency while maintaining reconstruction accuracy. Despite RENO's outstanding inference efficiency, it exhibits limitations in geometric compression accuracy, primarily manifested in insufficient structural representation capability and coarse contextual modeling granularity. On one hand, target-scale voxel features heavily rely on direct propagation from parent-scale features, lacking explicit characterization of relative structural positions and semantic contribution differences among key voxels. This hinders the model's ability to fully express local morphological variations in regions with intense geometric changes—such as boundary areas, thin structures, and sparse regions—thereby compromising the discriminative power and robustness of occupancy predictions. On the other hand, occupancy probability modeling inadequately decomposes and utilizes position-level dependencies, making it difficult to finely capture hierarchical conditional constraints between different segments. This further limits improvements in probability estimation accuracy and entropy encoding efficiency.

To address the aforementioned issues, this paper proposes a Structure-Aware and Context-Modeling Point Cloud Compression (SACM-PCC) method. Through the synergistic design of structure representation enhancement and bit-level probability modeling refinement, it simultaneously improves compression efficiency and reconstruction quality. To address insufficient feature representation capability, this paper designs a structure-aware feature embedding module. Building upon parent-scale feature priors, it introduces a voxel attention gating mechanism to adaptively re-weight the semantic contributions of different sub-voxels. This is combined with relative position embeddings generated by octree indexing to achieve cross-scale structural alignment and effective feature propagation between voxels. This enhances the model's ability to discern local geometric details and boundary structures. To address coarse contextual modeling granularity, we construct a bitwise progressive object occupancy predictor. This decomposes 8-bit occupancy codes in a biased manner from high to low bits and refines probability estimates incrementally using conditional autoregression. This enables lower-bit predictions to fully leverage decoded high-bit information, thereby improving bit-level discrimination accuracy and probability modeling stability. Furthermore, to enhance spatial context utilization efficiency, this paper introduces mutually independent sparse spatial context modeling modules at each prediction stage. Sparse 3D convolutions within the object voxel neighborhood aggregate local spatial information, strengthening the modeling capability of lateral correlations between voxels. This approach improves occupancy prediction consistency and entropy coding efficiency.

2. Related Work

2.1. Tree Structure

Point cloud compression is a critical component in the storage and transmission of 3D point cloud data, with its performance directly impacting the efficiency and quality of downstream perception and reconstruction applications. Tree structures, valued for their compact representation and clear hierarchical relationships, are widely employed for point cloud geometric encoding. Typically, kd-trees are integrated into systems like Google Draco [3] to support efficient real-time compression tasks; Octrees, meanwhile, recursively subdivide and describe local geometry through hierarchical partitioning of parent-child nodes, and are incorporated into the G-PCC standard proposed by MPEG [4]. G-PCC relies on heuristic rules for multi-level partitioning and reconstruction of point cloud geometry. Combined with context-adaptive probabilistic modeling and entropy coding, it demonstrates robust rate-distortion performance. However, compared to engineered implementations like Draco, it often lags in processing speed and real-time capability.

To further enhance compression efficiency and modeling capabilities, recent studies have introduced learnable context modeling mechanisms into tree-based representations. For instance, methods such as OctSqueeze [5] and OctAttention [6] enhance the expressive power of tree encoding by learnably modeling node/subnode states. Subsequent works like ECM-OPCC [7] and EHEM [8] further refined context modeling and entropy encoding strategies to improve encoding efficiency and reconstruction quality. However, these approaches typically still rely on a layer-by-layer generation and traversal process for tree structures. The encoding workflow incurs significant serial overhead during structure generation and traversal, creating inference bottlenecks that limit scalability and deployment efficiency in real-time applications.

2.2. Multiscale Sparse Tensor

Beyond tree-based representations, multiscale sparse tensors have emerged as a key technical approach for point cloud geometric compression in recent years. These methods typically utilize sparse tensors as a carrier, enhancing occupancy probability estimation accuracy through multiscale feature extraction and inter-scale information transfer, thereby enabling more refined geometric compression modeling. SparsePCGC [9] stands as a representative work in this field. Its core approach aggregates contextual information across scales through a multiscale sparse convolutional network, leveraging inter-scale structural correlations to enhance occupancy modeling capabilities. It demonstrates favorable rate-loss performance in both dense point cloud and sparse LiDAR point cloud tasks.

Despite these advances, multi-scale sparse tensor methods still face challenges in terms of efficiency and accuracy. Multi-stage, serialized inference processes often incur high computational overhead and inference latency, and can easily become performance bottlenecks when dealing with large-scale point clouds or complex scenes. To mitigate these issues, subsequent studies such as SparsePCCv2 [10] and Unicorn [11] have introduced improvements in both network architecture design and context modeling mechanisms. While these approaches have enhanced coding quality and modeling efficiency to some extent, the computational burden imposed by multi-stage inference remains difficult to eliminate entirely. Concurrently, other approaches have attempted to combine octree hierarchical partitioning with sparse tensor representations. For instance, [12,13] introduced hierarchical priors to reduce computational overhead while preserving expressive power. Nevertheless, the fundamental trade-off between complexity control and compression efficiency persists, as these methods still rely on the incremental generation process of hierarchical voxel refinement and occupancy prediction.

2.3. Range Images

Due to the physical characteristics of LiDAR sensors, LiDAR point clouds can be mapped onto two-dimensional range images through spherical or cylindrical projection, thereby transforming three-dimensional geometric modeling problems into two-dimensional image-based modeling and encoding. This representation enables efficient organization and encoding/decoding of dense geometric information on a two-dimensional plane, leading to the development of numerous range image-based compression methods [14,15,16]. Such methods typically leverage established two-dimensional image processing and compression techniques to perform feature modeling and entropy coding on range images. This approach reduces storage and transmission complexity while enhancing engineering implementation efficiency.

However, projection representation inevitably introduces information loss and geometric distortion. The projection process may result in the loss of information in occluded areas, compression of boundary details, and distortion of local structures, thereby affecting the accuracy and consistency of reconstructed point clouds at edges and in complex geometric regions. Particularly in high-precision point cloud compression tasks, when scenes contain sparse structures, slender objects, or strong occlusion relationships, 2D domain modeling often struggles to fully preserve the integrity and consistency of 3D spatial structures. This limitation restricts the applicability of such methods in high-fidelity geometric compression scenarios.

3. Method

As shown in Figure 1, SACM-PCC represents the input point cloud in a Cartesian coordinate system as a multiscale sparse voxel coordinate sequence $\{C^0, C^1, \ldots, C^D\}$, progressing from coarse to fine scales, where C^d denotes the sparse voxel coordinate set at scale d. This sequence is constructed through hierarchical binary subsampling, thereby reducing the additional computational overhead associated with hierarchical construction and traversal.

During the downsampling stage from dimension d to $d-1$, the Fast Occupancy Generator (FOG) aggregates voxel neighborhood information to generate an 8-bit sparse occupancy code for each parent voxel. This code describes the occupancy pattern of its 2×2×2 subblock:

$$(C^{d-1}, O^{d-1}) = \mathrm{FOG}(C^d), \quad d = D, \ldots, 1, \quad (1)$$

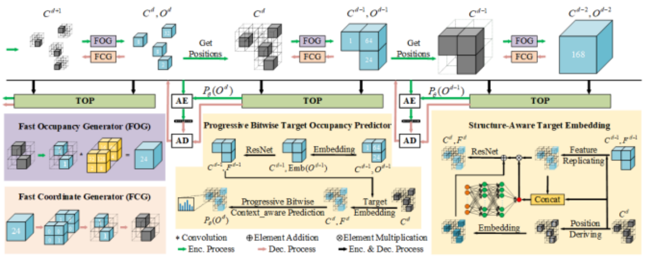

Figure 1. SACM-PCC Framework Overall Structure Diagram

Where $O^{d-1} = \{o_i^{d-1}\}_{i=1}^{N_{d-1}}$ corresponds one-to-one with the parent scale coordinate, $N_{d-1} = |C^{d-1}|$, and $o_i^{d-1} \in \{0,1,\ldots,255\}$ denotes the occupancy pattern of the parent voxel's sub-blocks. Subsequently, during the upscaling recovery phase from scale $d-1$ to d, the Fast Coordinate Generator (FCG) performs binary expansion on the occupancy code and prunes invalid sub-voxels to recover the target scale coordinate set:

$$C^d = \mathrm{FCG}(C^{d-1}, O^{d-1}), \quad d = 1,2,\ldots,D. \tag{2}$$

Therefore, given coarse-scale coordinates C^0 and a multi-scale occupancy code sequence $O = \{O^0, O^1, \ldots, O^{D-1}\}$, the decoder can progressively recover fine-scale geometry layer by layer and ultimately reconstruct the point cloud. Geometric compression can be equivalently transformed into efficient modeling and entropy coding of the occupancy code sequence O.

To probabilistically model O, this paper introduces a Structure-Aware Target Embedding (SATE) module and a Bitwise Progressive Target Occupancy Predictor (PBTOP) within a multi-scale sparse representation framework. Together, they construct a conditional probability quality function for occupancy codes. During scale-progressive modeling, occupancy codes at scale d are predicted conditionally on geometric priors from the preceding scale, defining structural prior features:

$$\Phi^{d-1} = \mathrm{SATE}(O^{d-1}, C^{d-1}) \tag{3}$$

the overall probability decomposes as:

$$P_\theta(O) = \prod_{d=1}^{D-1} P_\theta\left(O^d \mid O^{d-1}, C^{d-1}, \Phi^{d-1}\right) \tag{4}$$

Given the true distribution $P(O)$, the model undergoes maximum likelihood training, equivalent to minimizing the negative log-likelihood (cross-entropy) objective:

$$\theta^* = \arg\min_\theta \mathbb{E} \qquad \log P_\theta(O)\big] \tag{5}$$

This enhances occupancy probability estimation accuracy and further improves entropy coding efficiency.

3.1. Structure-Aware Object Embedding

To achieve efficient transfer and structural alignment of cross-scale voxel features, this paper proposes the Structure-Aware Target Embedding (SATE) module. The network structure diagram is shown in Figure 2. This module utilizes sparse occupancy codes and voxel features from the parent scale (depth $d-1$) as prior information. It first performs sparse 3D convolutions and residual aggregation at the parent scale to extract contextual features; Subsequently, based on the parent-child voxel correspondence, the parent voxel features are replicated/propagated to its 2×2×2 set of 8 child voxels, serving as initialization for the target scale (depth d) features, thereby enabling cross-scale information transfer.

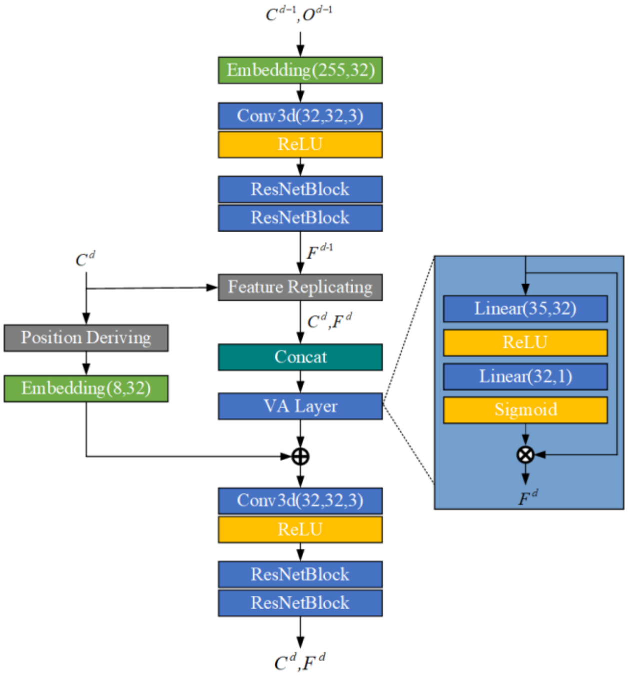

Figure 2. SATE Network structure Diagram

At the target scale, SATE further explicitly injects structural and positional information to enhance representational capabilities: On one hand, it introduces a voxel attention gating mechanism that combines the geometric coordinates of target voxels with initialized features to predict gating weights. This enables adaptive re-calibration of semantic contributions from different sub-voxels, highlighting key voxels while suppressing redundant ones. On the other hand, it constructs relative position embeddings using octant indices of sub-voxels within parent voxels, fusing them with attention-modulated features to explicitly encode the structural-positional relationship between “sub-voxels relative to parent voxels.” Finally, the module performs sparse 3D convolutions and residual refinement at the target scale, aggregating local neighborhood context to output structurally enhanced target-scale feature representations. By integrating cross-scale priors, attention gating, and relative position embeddings, SATE robustly captures local geometric variations in boundaries, thin structures, and sparse regions, providing more discriminative input representations for subsequent occupancy probability modeling and entropy encoding.

3.2. Progressive Bitwise Target Occupancy Predictor

To enhance the probabilistic modeling accuracy of target occupancy codes and improve spatial context adaptation, this paper proposes the Progressive Bitwise Target Occupancy Predictor (PBTOP). The network structure diagram is shown in Figure 3. At scale d, given the structure-aware feature F^d and the 8-bit occupancy code $O^d \in \{0,1,\ldots,255\}$ of a target voxel, PBTOP decomposes O^d into four subcodes from high to low bits. It progressively introduces the conditional constraint of “high-bit prior refinement to low-bit detail,” thereby reducing the classification space per iteration and improving estimation stability:

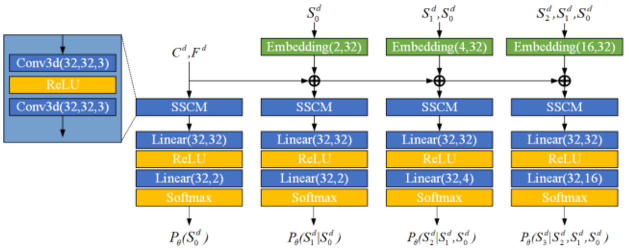

Figure 3. PBTOP Network structure Diagram

$$\begin{gathered} S_0^d \in \{0,1\}, \quad S_1^d \in \{0,1\}, \\ S_2^d \in \{0,1,2,3\}, \quad S_3^d \in \{0,1,\ldots,15\}, \end{gathered} \tag{6}$$

and satisfy the bit weight rearrangement relationship:

$$\hat{O}^d = 128S_0^d + 64S_1^d + 16S_2^d + S_3^d. \tag{7}$$

Based on the above decomposition, PBTOP employs conditional autoregressive modeling to factorize the occupancy code:

$$P_\theta(O^d) = \prod_{i=0}^{3} P_\theta\left(S_i^d \mid S_{<i}^d, \mathbf{C}^d, \mathbf{F}^d\right), \tag{8}$$

Here, $S_{<i}^d$ denotes the subcode results determined prior to stage i, $\mathbf{C}^d$ represents the target scale voxel coordinate set, and $\mathbf{F}^d$ denotes the structure-aware features output by SATE. This factorized form enables lower-level predictions to explicitly leverage decoded higher-level information, thereby enhancing probability estimation accuracy and facilitating entropy coding.

In its implementation, PBTOP comprises four stages $i \in \{0,1,2,3\}$,each stage first extracts phased contextual features within a local sparse neighborhood through Sparse Spatial Context Modeling (SSCM):

$$\mathbf{F}_i^d = \mathrm{Conv3D}_{\mathrm{sp}}^{(2)}\left(\mathrm{ReLU}\left(\mathrm{Conv3D}_{\mathrm{sp}}^{(1)}(\mathbf{F}^d)\right)\right) \tag{9}$$

The known subcode conditions are then encoded into stage condition vectors and fused with $\mathbf{F}_i^d$ to form the classification input. Finally, a stage-specific MLP outputs the conditional probability distribution for the corresponding category, enabling stage-by-stage prediction of S_i^d . Through the combined design of "high-bit prior-driven, low-bit incremental refinement" and "local sparse context aggregation," PBTOP achieves finer-grained characterization of bit-level dependencies and spatial correlations, thereby reducing the expected code length and enhancing overall compression efficiency.

3.3. Loss Function

To learn the conditional probability distribution of the multi-scale sparse occupancy sequence $O = \{O^0, O^1, \ldots, O^{D-1}\}$ and enhance entropy coding efficiency, this paper employs cross-entropy loss (negative log-likelihood) as the training objective. Let the true distribution be $P(O)$ and the joint probability mass function predicted by the model be $P_\theta(O)$. The overall loss is defined as:

$$\mathcal{L} \qquad \log P_\theta(O)\Big] \tag{10}$$

Among these, $P_\theta(O)$ is modeled using scale-progressive conditions; within each scale, the 8-bit occupancy code $O^d \in \{0,1,\ldots,255\}$ is decomposed into four-stage subcodes from high to low bits. Their probabilities are estimated sequentially using conditional autoregressive methods, aligning training with sequential decoding and entropy coding processes. This enhances probability modeling accuracy while reducing the expected code length.

4. Experiments and Discussions

4.1. Dataset and Experimental Environment

To comprehensively evaluate the overall performance of the proposed method in point cloud compression tasks, this paper selects two representative public LiDAR point cloud datasets, KITTI [17]and Ford [18], for experimentation. The KITTI dataset, acquired using a Velodyne HDL-64E, contains approximately 14,999 point cloud frames. Following the official partitioning, 3,712 frames are used for training, 3,769 for validation, and 7,518 for testing. The Ford dataset, collected by LiDAR devices and constructed according to the MPEG point cloud compression evaluation specification, contains 1,500 point cloud frames. This paper employs a standardized testing setup: training is performed on sequence 01, while sequences 02 and 03 are used for testing and evaluation to ensure fairness and reproducibility of results. Model implementation utilizes Python 3.10 and PyTorch 2.0, integrating TorchSparse++ to accelerate sparse 3D convolutions; Training employs the Adam optimizer with a multi-step learning rate strategy, initializing at $5\mathrm{e}-4$, with a batch size of 1 and 150,000 total iterations. Inputs undergo FP16 quantization to reduce GPU memory consumption. All experiments were conducted on a server platform equipped with an NVIDIA GeForce RTX 4090D (24GB VRAM).

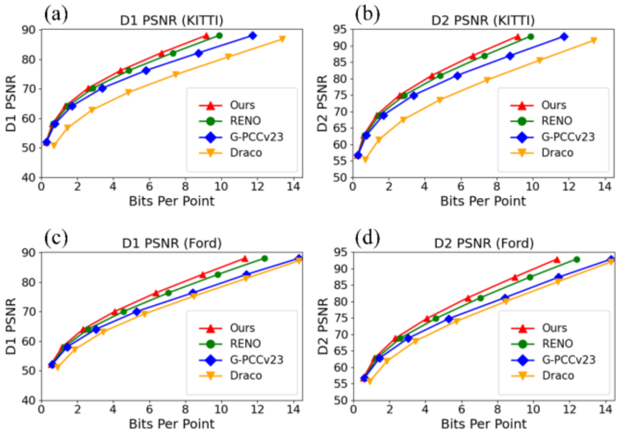

Figure 4. Bitrate-Distortion Performance Comparison on KITTI and Ford

4.2. Comparative Studies

To validate the effectiveness of the proposed method, this paper evaluates it on two mainstream point cloud datasets, KITTI and Ford, and compares it with three representative compression methods: the traditional standard method G-PCCv23, the industrial tool Draco, and the learning-based neural compression method RENO. Evaluation metrics include D1-PSNR, D2-PSNR, and bit rate (BPP), with overall compression efficiency further measured by BD-BR. As shown in Figures 4(a)–(d), the proposed method (Ours) consistently outperforms the comparison methods on the D1/D2 rate-distortion curves for both KITTI and Ford. This indicates that higher PSNR can be achieved at the same BPP, with the advantage becoming more pronounced in the medium-to-high bitrate range. Furthermore, as shown in Table 1, SACM-PCC achieves significant bitrate savings over all comparison methods on the BD-BR metric. On the KITTI dataset, it achieves approximately 57% bitrate savings relative to Draco (D1: -57.12%, D2: -57.08%), approximately 21% relative to G-PCCv23 (D1: -21.04%, D2: -21.01%), and 8.7% relative to RENO (D1: -8.78%, D2: -8.75%). On the Ford dataset, the relative bitrate savings compared to Draco are approximately 54% (D1: -54.11%, D2: -54.07%), approximately 21%–22% relative to G-PCCv23 (D1: -21.60%, D2: -21.79%), and 9% relative to RENO (D1: -9.1%, D2: -9.08%). The above results consistently demonstrate that the proposed method significantly reduces bit overhead while maintaining reconstruction quality on both datasets. Compared to G-PCCv23 and Draco, it exhibits superior compression efficiency, while outperforming RENO in terms of more stable and consistent D1/D2 metrics.

Table 1. Quantitative Gain of SACM-PCC Compared to Other Methods for BD-BR

Dataset	Metric	Draco	G-PCC	RENO	Ours
KITTI	D1(%)	-57.12	-21.04	-8.78	-
	D2(%)	-57.08	-21.01	-8.75	-
Ford	D1(%)	-54.11	-21.60	-9.10	-
	D2(%)	-54.07	-21.79	-9.08	-

4.3. Qualitative Visualization

As shown in Figure 5, this figure visually compares different compression methods under low bitrate and high-fidelity reconstruction scenarios. Taking two typical KITTI data frames as examples, it is evident that the proposed method achieves reconstruction quality comparable to current state-of-the-art approaches while maintaining a lower encoding bitrate, demonstrating superior compression efficiency and overall performance. In contrast, while the RENO method maintains high structural fidelity, it requires relatively high bitrates, limiting its compression efficiency. Traditional methods like G-PCC and Draco exhibit noticeable blurring and information loss in boundary and detail regions, resulting in suboptimal overall reconstruction quality alongside relatively higher compression overhead.

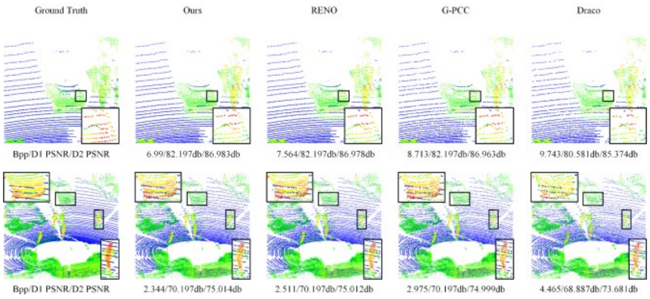

Figure 5. Visualization Results of Point Cloud Reconstruction Using Different Methods

4.4. Ablation Studies

As shown in Table 2, both SATE and PBTOP deliver stable improvements in BD-BR over RENO, with the combined approach yielding the optimal results. When SATE is added alone, D1/D2 on KITTI increase by 1.42%/1.40%, and on Ford by 1.24%/1.22%, indicating its structural modeling brings gains. The improvement from adding PBTOP alone is even more pronounced, reaching 7.33%/7.31% on KITTI and 7.82%/7.79% on Ford, highlighting the critical role of bitwise progressive prediction in probability modeling. The combined approach achieved the best results: 8.78%/8.75% on KITTI and 9.10%/9.08% on Ford, demonstrating strong complementarity and synergistic gains between structural representation and probabilistic prediction.

Table 2. Quantitative Gain of BD-BR Relative to RENO for Each Module

Modules	KITTI D1(%)	KITTI D2(%)	Ford D1(%)	Ford D2(%)
RENO	-	-	-	-
RENO+SATE	1.42	1.40	1.24	1.22
RENO+ PBTOP	7.33	7.31	7.82	7.79
RENO+SATE+PBTOP	8.78	8.75	9.10	9.08

5. Conclusion

This paper proposes a point cloud compression method based on structure-aware and context-aware modeling, comprising a structure-aware object embedding module and a bitwise progressive object occupancy predictor. These components respectively enhance cross-scale structural alignment and feature propagation capabilities, while improving the probability prediction accuracy of sparse occupancy codes and context modeling effectiveness. Experimental results on the KITTI and Ford public point cloud datasets demonstrate that this method outperforms G-PCC, Draco, and RENO in both compression rate (Bits Per Point) and reconstruction accuracy (D1/D2 PSNR). Future research will focus on multi-frame collaborative modeling, cross-scenario generalization, and adaptive bitrate control to enhance practical deployment performance.

References

[1] Roriz R, Silva H, Dias F, et al. A survey on data compression techniques for automotive lidar point clouds[J]. Sensors, 2024, 24(10): 3185.

[2] You K, Chen T, Ding D, et al. Reno: Real-time neural compression for 3d lidar point clouds[C]//Proceedings of the Computer Vision and Pattern Recognition Conference. 2025: 22172-22181.

[3] Galligan F, Hemmer M, Stava O, et al. Google/draco: a library for compressing and decompressing 3d geometric meshes and point clouds[J]. Draco: a library for compressing and decompressing 3D geometric meshes and point clouds, 2018.

[4] Schwarz S, Preda M, Baroncini V, et al. Emerging MPEG standards for point cloud compression[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2019, 9(1): 133-148.

[5] Huang L, Wang S, Wong K, et al. Octsqueeze: Octree-structured entropy model for lidar compression[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1313-1323.

[6] Fu C, Li G, Song R, et al. Octattention: Octree-based large-scale contexts model for point cloud compression[C]//Proceedings of the AAAI conference on artificial intelligence. 2022, 36(1): 625-633.

[7] Jin Y, Zhu Z, Xu T, et al. Ecm-opcc: Efficient context model for octree-based point cloud compression[C]//ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2024: 7985-7989.

[8] Song R, Fu C, Liu S, et al. Efficient hierarchical entropy model for learned point cloud compression[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023: 14368-14377.

[9] Wang J, Ding D, Li Z, et al. Sparse tensor-based multiscale representation for point cloud geometry compression[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(7): 9055-9071.

[10] Xue R, Wang J, Ma Z. Efficient LiDAR point cloud geometry compression through neighborhood point attention[J]. arXiv preprint arXiv:2208.12573, 2022..

[11] Wang J, Xue R, Li J, et al. A versatile point cloud compressor using universal multiscale conditional coding–Part I: Geometry[J]. IEEE transactions on pattern analysis and machine intelligence, 2024.

[12] Fan T, Gao L, Xu Y, et al. Multiscale latent-guided entropy model for lidar point cloud compression[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 33(12): 7857-7869.

[13] Lodhi M A, Pang J, Tian D. Sparse convolution based octree feature propagation for lidar point cloud compression[C]//ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023: 1-5.

[14] Stathoulopoulos N, Saucedo M A V, Koval A, et al. RecNet: An invertible point cloud encoding through range image embeddings for multi-robot map sharing and reconstruction[C]//2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024: 4883-4889.

[15] Wang S, Jiao J, Cai P, et al. R-pcc: A baseline for range image-based point cloud compression[C]//2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022: 10055-10061.

[16] Zhou X, Qi C R, Zhou Y, et al. Riddle: Lidar data compression with range image deep delta encoding[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022: 17212-17221.

[17] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite[C]//2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012: 3354-3361.

[18] Pandey G, McBride J R, Eustice R M. Ford campus vision and lidar data set[J]. The International Journal of Robotics Research, 2011, 30(13): 1543-1552.