UniPCGC: Towards Practical Point Cloud Geometry Compression via an Efficient Unified Approach

Kangli Wang[1], **Wei Gao**[1,2*]
[1]Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, Shenzhen Graduate School, Peking University, Shenzhen 518055, China
[2]Peng Cheng Laboratory, Shenzhen, China
kangliwang@stu.pku.edu.cn, gaowei262@pku.edu.cn

Abstract

Learning-based point cloud compression methods have made significant progress in terms of performance. However, these methods still encounter challenges including high complexity, limited compression modes, and a lack of support for variable rate, which restrict the practical application of these methods. In order to promote the development of practical point cloud compression, we propose an efficient unified point cloud geometry compression framework, dubbed as UniPCGC. It is a lightweight framework that supports lossy compression, lossless compression, variable rate and variable complexity. First, we introduce the Uneven 8-Stage Lossless Coder (UELC) in the lossless mode, which allocates more computational complexity to groups with higher coding difficulty, and merges groups with lower coding difficulty. Second, Variable Rate and Complexity Module (VRCM) is achieved in the lossy mode through joint adoption of a rate modulation module and dynamic sparse convolution. Finally, through the dynamic combination of UELC and VRCM, we achieve lossy compression, lossless compression, variable rate and complexity within a unified framework. Compared to the previous state-of-the-art method, our method achieves a compression ratio (CR) gain of 8.1% on lossless compression, and a Bjontegaard Delta Rate (BD-Rate) gain of 14.02% on lossy compression, while also supporting variable rate and variable complexity.

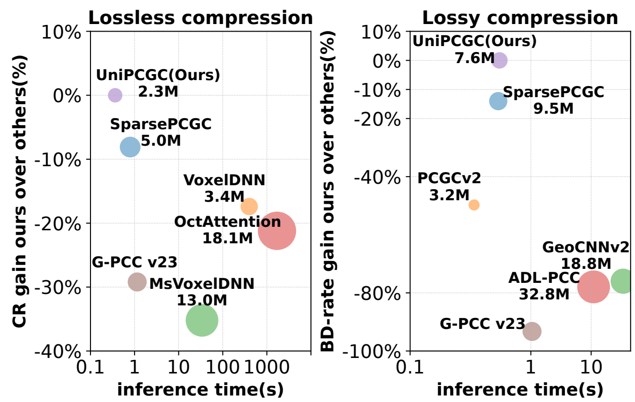

Figure 1: Performance comparisons for the proposed method and the other methods. The left figure shows the CR gain obtained by the proposed UniPCGC method in lossless compression. The right figure shows the BD-rate gain obtained by the proposed UniPCGC method in lossy compression.

Introduction

3D point clouds have been widely used in applications such as immersive media communication and digital culture heritage, etc. There has been an increasing amount of point cloud data generated. To relieve the burden of massive data transmission and storage, there is an immediate requirement to develop point cloud compression algorithms with high efficiency. Although non-learning compression algorithm such as Geometry based Point Cloud Compression (Cao et al. 2021) (G-PCC) has been developed for point cloud compression, its performances is still not satisfactory. With the advancement of deep learning, numerous methods base on learning have been proposed. However, current learning-based methods still need to be improved in terms of performance and flexibility.

In point cloud compression, accurate probability estimation of point occupancy state can significantly reduce the bit rate. Some methods (Nguyen et al. 2021a,b; Fu et al. 2022; Nguyen and Kaup 2023) achieve satisfactory performance by adopting autoregressive approaches for probability estimation. However, the decoding complexity of autoregressive-based methods is prohibitively high, making them impractical for real-world applications. Many efforts (Wang et al. 2022; Song et al. 2023; Jin et al. 2024) focus on exploring more efficient grouping strategies to trade-off the performance and decoding complexity. However, the first group lacks any context information, which leads an unacceptably high bit rate. Reducing the bit rate of preceding groups becomes crucial for improving compression performance. To address this issue, we propose an uneven grouping scheme that reduces the size of the first group, allowing more points to benefit from context information.

Flexibility is crucial for point cloud compression algorithms. In terms of flexibility, (Wang et al. 2022) enhances flexibility by simultaneously supporting both lossy and lossless compression. In the field of image compression, certain lossy compression methods (Yin et al. 2022; Cui et al. 2021)

*Corresponding Author: Wei Gao.

have achieved variable rate through modulation techniques. However, the paradigm may result in decreased compression performance under a wide range of variable rates. Overall, most existing works fall short in terms of supporting flexibility, highlighting the urgent need for a sufficiently flexible point cloud compression algorithm. To address the issue, we propose channel-level rate modulation and dynamic sparse convolution, which enable variable rate and complexity. The approach significantly enhances the flexibility and applicability of point cloud compression.

To address these limitations, we present an efficient unified framework named UniPCGC. In the lossless mode, we propose a method named Uneven 8-Stage Lossless Coder (UELC). In the lossy mode, Variable Rate and Complexity Module (VRCM) are achieved. By dynamically combining UELC and VRCM, we establish the UniPCGC framework. It is a lightweight framework that supports lossy compression, lossless compression, variable rate and variable complexity. As shown in Figure 1, we obtain state-of-the-art performance in both lossless and lossy mode. Moreover, our model exhibits competitive model size and inference speed. Our contributions are summarized as follows:

- We propose an efficient unified point cloud compression framework UniPCGC, which is a lightweight framework that supports lossy compression, lossless compression, variable rate and variable complexity. To our best knowledge, this is the first unified compression framework that simultaneously support these four modes.
- To improve the efficiency of context grouping for better compression performances, we devise a method named Uneven 8-Stage Lossless Coder (UELC), which allocates more computational complexity to groups with higher coding difficulty, and merges groups with lower coding difficulty in the lossless mode.
- To enhance the flexibility and applicability, we provide the Variable Rate and Complexity Module (VRCM), which are designed for lossy compression through the proposed joint adoption of a rate modulation scheme and dynamic sparse convolution.
- Experimental results demonstrate that our method obtains state-of-the-art (SOTA) performance on both lossy and lossless compression, and supports variable rate and complexity, which facilitates practical applications.

Related Works

Point Cloud Geometry Compression

Voxel-Based Representation. In the exploration of point cloud compression, various approaches have been proposed, including those based on 3D convolution neural networks. PCGCv1 (Wang et al. 2019) has shown promising rate-distortion efficiency. To achieve efficient and low-complexity compression, PCGCv2 (Wang et al. 2021) employs sparse convolution (Choy, Gwak, and Savarese 2019) instead of traditional dense convolution. To utilize cross-scale information more efficiently, SparsePCGC has been proposed for both lossless and lossy compression, which estimates occupancy probabilities using cross-scale correlations in a single-stage or multi-stage manner (Wang et al. 2022). While these methods effectively maintain a relatively low level of complexity, their lack of support for variable rate and variable complexity limits their flexibility.

Octree-Based Representation. Octsqueeze (Huang et al. 2020) and Muscle (Biswas et al. 2020) adopt MLP for occupancy probability prediction. OctAttention (Fu et al. 2022) uses an octree to represent point clouds to reduce spatial redundancy, and then designs a conditional entropy model to model ancestor and sibling contexts to exploit the strong dependencies between adjacent nodes. EHEM (Song et al. 2023) resolves the issue of high decoding complexity by performing binary grouping of sibling nodes. Indeed, these methods encounter challenges in achieving satisfactory lossy compression performance on dense point clouds and fail to incorporate support for variable rate.

Hybrid-Based Representation. VoxelDNN (Nguyen et al. 2021a) adaptively partitions point clouds into multi-resolution voxel blocks based on their structural characteristics and represents the partitioning using an octree. SparseVoxelDNN (Nguyen and Kaup 2022) replaces 3D convolution with 3D sparse convolution and achieves better performance based on VoxelDNN. Under the autoregressive framework, the decoding complexity will become very high. Such point cloud compression methods are difficult to apply in practical scenarios due to their high decoding complexity. GRASP-Net (Pang, Lodhi, and Tian 2022) is another hybrid model framework. Although GRASP-Net maintains low complexity and achieves high performance, it only supports lossy compression and lacks support for variable rate, making it less practical in specific situations.

Variable Rate Support

λ is also used to adjust the distortion weight and control the target rate, which means that different models need to be trained with different λ values for different rate points, limiting the flexibility and generality of compression algorithms. To solve the problem, some methods use a single model to obtain variable rates. Conditional autoencoder schemes (Choi, El-Khamy, and Lee 2019; Yang et al. 2020; Sun et al. 2021; Song, Choi, and Han 2021; Lin et al. 2021) take prior information of λ as input to obtain variable rates. (Theis et al. 2016; Cui et al. 2021; Yin et al. 2022; Jia et al. 2022) define scaling factors or sub-networks to modify the magnitude of the latent representation, thereby obtaining variable rate through modulation. However, these works are primarily proposed in the field of image compression, and there is a scarcity of specifically tailored variable rate solutions available for point cloud compression.

Dynamic Neural Networks

Dynamic networks utilizes a per-sample conditioned dynamic architecture for inference. It not only reduces redundant computations for simple samples but also maintains their representational capacity when dealing with challenging samples. Reducing redundant computations can be achieved through dynamic depth, which include early exiting (Huang et al. 2017; Bolukbasi et al. 2017) and skip con-

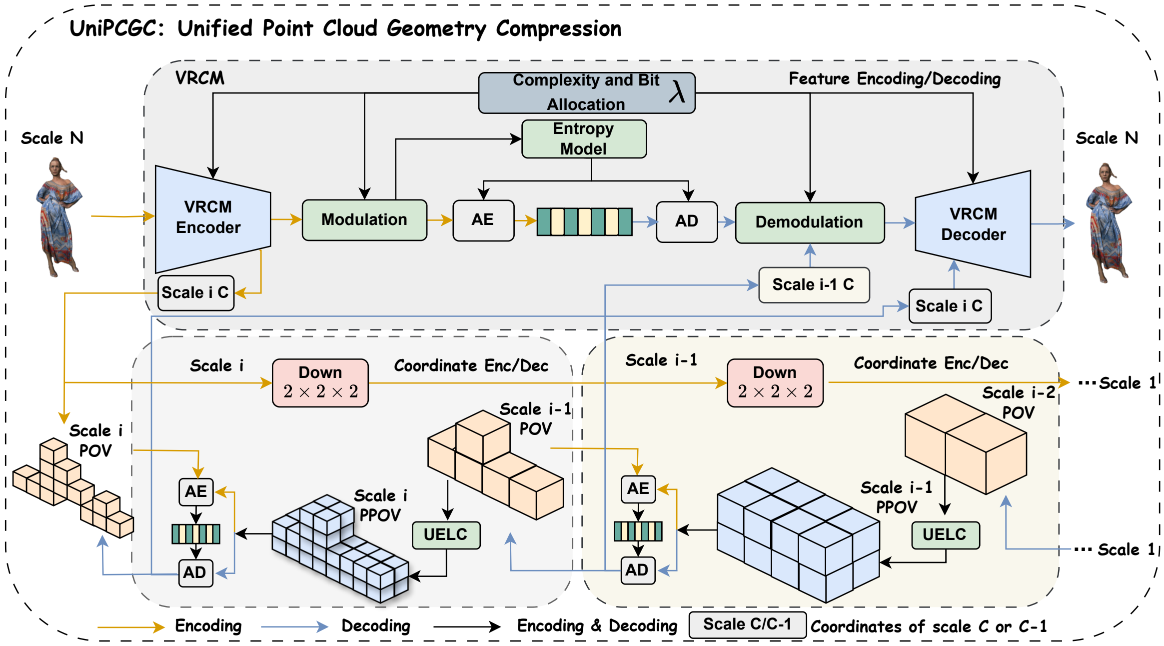

Figure 2: llustration of the proposed UniPCGC framework. It mainly consists of two parts: coordinate coding and feature coding. Coordinate coding is performed using Uneven 8-Stage Lossless Code (UELC) at each scale. Feature coding is performed using Variable Rate and Complexity Module (VRCM), which mainly includes encoder, modulation, demodulation, decoder, complexity and bit allocation and Factorized Entropy Model. AE/AD stands for arithmetic encoding and arithmetic decoding.

nections (Graves 2016; Wang et al. 2018; Veit and Belongie 2018). Dynamic width (Liu et al. 2017; Huang et al. 2018; Tao and Gao 2021) refers to executing every layer and selectively skipping certain redundant channels to reduce computational complexity in a more fine-grained manner. Dynamic neural networks have been introduced in the field of image compression (Tao et al. 2023), enabling control the encoding complexity while achieving 40% speedup. However, there is a lack of available dynamic networks solutions specifically tailored for point cloud compression.

From the above literature review, we can see that although several methods have shown satisfactory performance, practical point cloud compression should ideally support lossless compression, lossy compression, variable rate and complexity simultaneously. It is noted that existing methods often consider only one or two of these aspects, which limits their applicability. To the best of our knowledge, the proposed UniPCGC is the first solution that achieves the unified support while obtaining superior compression performances.

Method

Overview

The framework of the proposed UniPCGC is shown in Figure 2. UniPCGC mainly consists of two parts: coordinate coding and feature coding. In the lossless phase, the proposed Uneven 8-Stage Lossless Coder (UELC) is used. In the lossy phase, the proposed Variable Rate and Complexity Module (VRCM) is employed. UniPCGC enables lossless geometry compression by only using UELC for coordinate coding, while using UELC and setting dynamic downsampling times in VRCM allows lossy geometry compression at different rate ranges.

Notations and Blocks

Basic Notations. In our study, we represent point clouds using a voxelized approach. We denote a voxel as Positively or Non Occupied Voxel (POV or NOV) to indicate that the voxel is occupied or not occupied. PPOV represents Probable Positively Occupied Voxel. We utilize sparse convolution to extract features, where k refers to the kernel size, s refers to the stride, and c refers to the number of channels.

Feature Extraction Layer (FEL). We utilize the FEL for feature extraction. For dense object point clouds, sparse convolution with $k = 3$ has proven effective in feature extraction. However, the distances within the groups increase with finer grouping. To address this, we expand the receptive field by setting $k = 5$ to extract features. In FEL, we employ stacked Inception ResNet (IRN) for feature extraction.

Upsampling Layer (USL). In the layer, we apply a transposed convolution with $k = 2$ and $s = 2$.

Occupancy Probability Generation (OPG). The layer is located after the FEL and utilized to calculate the occupancy probability for each voxel. The proposed FEL, IRN, USL and OPG are illustrated in the Supplementary Material.

One-Stage Lossless Coder (OLC). OLC is commonly used for cross-scale probability estimation, utilizing the prior information of POV from $N - 1$ scale to directly upsample to the N scale in a single-stage. The detailed structure of OLC is illustrated in the Figure 4.

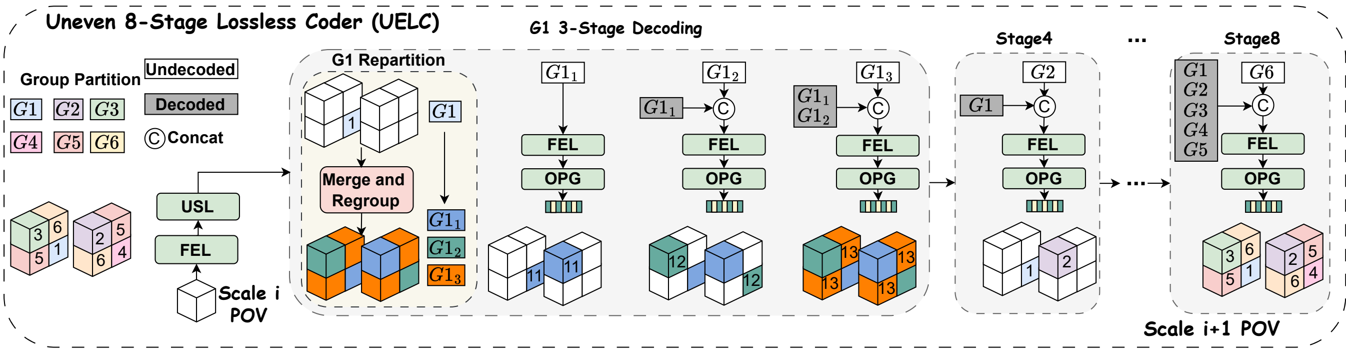

Figure 3: Illustration of the proposed Uneven 8-Stage Lossless Coder (UELC). In each stage, the previously coding groups and their features are regarded as prior information to better estimate the occupancy probability of groups in the current stage.

Uneven 8-Stage Lossless Coder (UELC)

In SparsePCGC, a uniform eight-stage approach is proposed, which achieves SOTA performance in lossless geometry compression. Upon re-evaluating this grouping method, we observe that the first group lacks any reference information, leading to an excessive number of bits required for encoding. However, in the later groups, more refined grouping brings little benefit. Therefore, we believe that more coding stages should be allocated in the first group, and subsequent groups should be combined. Based on this motivation, we propose the Uneven 8-stage Lossless Coder (UELC). In lossless compression, UELC mainly performs the probability prediction of PPOV. As shown in Figure 2, the more accurate the probability of PPOV estimation after UELC, the smaller bits required for arithmetic encoding. At the decoding end, arithmetic decoding can be used to restore the current scale POV. Figure 3 details the UELC step by step:

(1) First, the POV of Scale i is upsampled through USL and FEL to obtain the PPOV, which is divided into 6 groups as shown in the Figure 3. The processing within the group is parallel, and the processing between the groups is serial. After all processing is completed, we obtain the prediction of the occupancy probability of scale i+1.

(2) When encoding or decoding $G1$, we group $G1$ more precisely and divide $G1$ into $G1_1$, $G1_2$, and $G1_3$, marked as 11, 12, and 13 in Figure 3. Three groups are processed sequentially and complete in three stages. For each stage, we process the PPOV using stacked FEL and OPG blocks to determine its p_{PPOV} and compress the occupancy probability into the bitstream in the encoder or parse the bitstream in the decoder to identify the POV and NOV.

(3) For the current encoding or decoding stage, we prune the PPOV of the previous stage according to the POV, and use the pruned PPOV as prior information to assist in predicting the occupancy probability of the current stage.

(4) The subsequent stages are similar to the previous stage. We gradually complete the probability estimation for the entire scale using the proposed uneven 8-stage approach. The entire coordinate coding process can be expressed by the following formula:

$$\begin{aligned} \mathrm{bin}_i &= f_E\left(C_{i-1}, C_i\right) = \left(\mathrm{AE}\left(f_U\left(C_{i-1}\right), C_i\right)\right), \\ C_i &= f_D\left(C_{i-1}, \mathrm{bin}_i\right) = \left(\mathrm{AD}\left(f_U\left(C_{i-1}\right), bin_i\right)\right), \end{aligned} \tag{1}$$

where $f_U(\cdot)$ is the UELC, C_i represents the coordinates of scale i, and bin_i is the compressed file of scale i. In addition, the proposed UELC employs a non-sequential grouping scheme. Since UELC requires merging certain groups for joint encoding, it is essential to ensure that these merged groups can reference information from closer points as much as possible. Inspired by checkerboard context (He et al. 2021), we propose a non-sequential grouping scheme, as shown in Figure 3. At the m-th scale, we can estimate the bit rate of PPOV occupancy status:

$$R_{\mathrm{S}}(m) = \sum_k -\log_2\left(p^m_{\mathrm{PPOV}}(k)\right), \tag{2}$$

where $p^m_{\mathrm{PPOV}}(k)$ is the probability estimated by UELC. The aforementioned process involves occupancy probability estimation between two scales. We apply the proposed UELC from the lowest scale to the highest scale, ensuring that each scale shares the same weights, which enhances the flexibility of the model.

Variable Rate and Complexity Support

In this section, we propose the Variable Rate and Complexity Module (VRCM) for lossy geometry compression to support variable rate and variable complexity. The Figure 4 illustrates the main framework of VRCM. VRCM completes the feature encoding part, and the coordinate C_i encoding is completed by UELC. Therefore, at the decoding end, C_i and C_{i-1} are prior information.

At the encoder, different downsampling times **(0, 1, 2)** are performed according to the bit rate range (high, medium, low). It can be represented by $f^n_{down}(\cdot)$, where n is the number of downsampling times. Subsequently, the latent feature y is obtained by passing through the FEL, Down, DFEL, Down, FEL layers respectively. After passing through the first Down layer, the coordinates of this scale C_i are sent to the coordinate encoding part for lossless encoding. The above process can be represented by $f_{enc}(x, \lambda)$. Among them, λ is the complexity and rate modulation factor, which modulates the computational complexity of the module at the encoder and decoder. Moreover, λ generates the channel-level rate modulation coefficient through channel level bit allocation module, which is modulated with y at the encoder

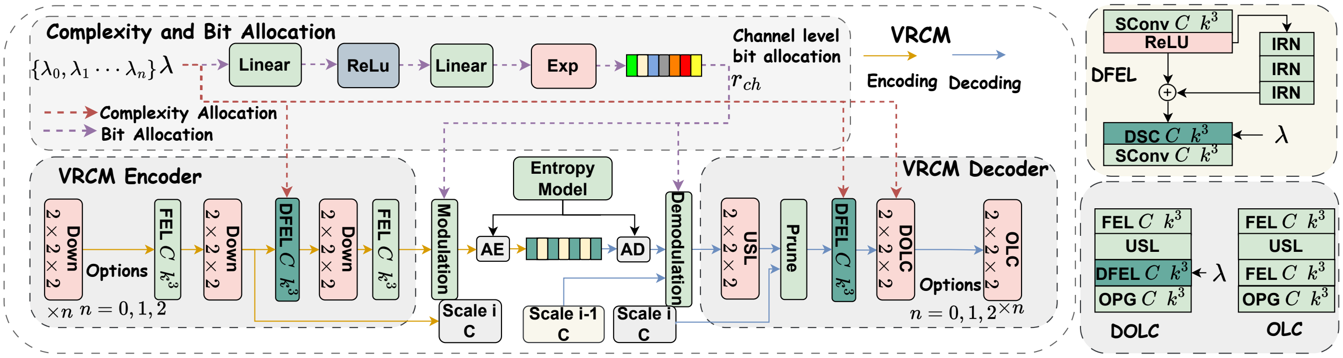

Figure 4: Detailed architecture of proposed VRCM. It also shows the architecture of channel level bit allocation module, One-Stage Lossless Coder (OLC / DOLC) and Dynamic Feature Extraction Layer (DFEL).

and demodulated by $\hat{y}_m$ at the decoder. At the decoder, the demodulated features are combined with C_{i-1} to form a sparse tensor. After USL, the correct coordinates C_i will be used to crop the wrong voxels to reduce distortion. Finally, it passes through DFEL, DOLC and $f_{up}^{n}(\cdot)$. The above process can be expressed by the following formula:

$$\begin{gathered} y = f_{enc}\left(f_{down}^{n}\left(p_{ori}\right), \lambda\right), \\ r_{ch} = f_{Exp}\left(f_L\left(f_R\left(f_L(\lambda)\right)\right)\right), \\ y_m = y * r_{ch}, \hat{y}_m = Q\left(y_m\right), y_{dm} = \hat{y}_m / r_{ch}, \\ \hat{y} = Cat\left(y_{dm}, C_{i-1}\right), \\ p_{res} = f_{up}^{n}\left(f_{dec}\left(\hat{y}, \lambda, C_i\right)\right), \end{gathered} \tag{3}$$

where p_{ori} is the original input point cloud, $f_L(\cdot)$ is the linear layer, $f_R(\cdot)$ is the Relu activation function, $f_{Exp}(\cdot)$ is the exponential function, r_{ch} is the channel-level rate modulation coefficient, C_i represents the coordinate of scale i, and $Cat(\cdot)$ means merging the input to generate a sparse tensor. The value of $R_{y_m}(i)$ can be calculated as follows:

$$R_{y_m}(i) = \sum_j -\log_2\left(p_{R_{y_m}^i}(j)\right), \tag{4}$$

where $p_{R_{y_m}^i}(j)$ follows the factorized entropy model.

Moreover, in the works of PCGCv2 and SparsePCGC, the factorized entropy model (Ballé et al. 2018) is utilized. It assumes that there is no statistical dependency within the probability distribution. Hence, the essence of compression lies in removing statistical dependencies as much as possible during the encoding process (Ali et al. 2024). Inspired by this, we propose dynamic sparse convolution (DSC), where points with high correlation are allocated more computational resources to explicitly remove statistical dependencies, thereby maintaining compression performance while reducing encoding complexity. We can define the correlation between the current point and surrounding points as:

$$Corr = E_{x\sim p(x)}[(\frac{x_i - \mu_i}{\sigma_i})(\frac{x_c - \mu_c}{\sigma_c})], \tag{5}$$

where $0 \le i < k^2$, c represents the current point, and μ and σ represent the mean and variance of the feature vector of the points. Next, we divide the points in the space into two groups based on the correlation. The specific calculation definition is as follows:

$$\begin{gathered} x_1, x_2 = f_{spilt}\left(f_{conv}(x), \lambda \mid Corr\right), \\ out = f_{conv}\left(x_1\right) \cup x_2, \end{gathered} \tag{6}$$

where $f_{spilt}(\cdot)$ is the split operation, $f_{conv}(\cdot)$ is the sparse convolution layer, and $\cup$ is the concatenation of two sparse tensors. Next, we replace the convolutions in OLC and FEL with DSC, resulting in Dynamic One-Stage Lossless Coder (DOLC) and Dynamic Feature Extraction Layer (DFEL), as shown in the Figure 4. The above is the main introduction of VRCM. Next, we discuss the training methodology for VRCM. We propose a multi-stage training strategy for VRCM. **Stage1:** We set the λ value of the loss function to 0.1 to achieve fast optimized. **Stage2:** We set the λ value to 1.0 for stable optimized. **Stage3:** We train multiple loss functions with rate control factors λ to achieve variable rate and variable complexity, as shown in equation (9).

Loss Functions

We use Binary Cross Entropy (BCE) in training to estimate the difference between predicted occupancy probabilities and actual occupancy labels:

$$L_{BCE} = \sum_k -(o_1(k)\log_2 p_1(k) + o_2(k)\log_2 p_2(k)), \tag{7}$$

where $o_1(k)$ and $o_2(k)$ represent the actual occupancy and non-occupancy symbol respectively, p_1 and p_2 are the probability that the k-th PPOV is POV or NOV. In lossless compression, we train UELC on the highest four scales. In lossy compression, we train VRCM with n set to 0, 1, and 2. Additionally, the loss function must incorporate the rate consumption of features, as shown in the following equation:

$$L_{all} = \lambda L_{BCE} + R_{y_m}, \tag{8}$$

where R_{y_m} is calculated by equation (4) and the above equation is used in the first and second stages of training in lossy compression. In the third stage, we train multiple loss functions with rate control factors λ to achieve variable rate and variable complexity as follows:

$$L_{all} = \sum_{\lambda \in T}[R(\hat{y};\theta,\lambda) + \lambda L_{BCE}(x,\hat{x};\theta,\lambda)], \tag{9}$$

Point clouds	Ours	GPCC	SparsePCGC	VoxelDNN	MsVoxelDNN	OctAttention	ECM-OPCC
Red&black × 300	0.59	0.82	0.64	0.67	0.87	0.73	0.66
Loot × 300	0.49	0.69	0.53	0.58	0.73	0.62	0.55
Thaidancer × 1	0.51	0.70	0.56	0.68	0.85	0.65	0.58
Boxer × 1	0.45	0.65	0.49	0.55	0.70	0.59	0.51
Average Bpp ↓	**0.51**	0.72	0.56	0.62	0.79	0.65	0.58
Ours CR Gain ↑	**0.0%**	-29.2%	-8.1%	-17.4%	-35.2%	-21.2%	-11.3%
Enc time(s) ↓	*0.56	*1.99	*0.71	885	54	1.06	1.92
Dec time(s) ↓	*0.57	*1.49	*0.88	640	58	1229	19.5

Table 1: Performance of lossless methods on the 8iVFB testset under the same training conditions. UniPCGC, GPCC v23 and SparsePCGC are tested using RTX 4080 GPU and intel i5-13600KF CPU for a fair runtime comparison (Mark with *).

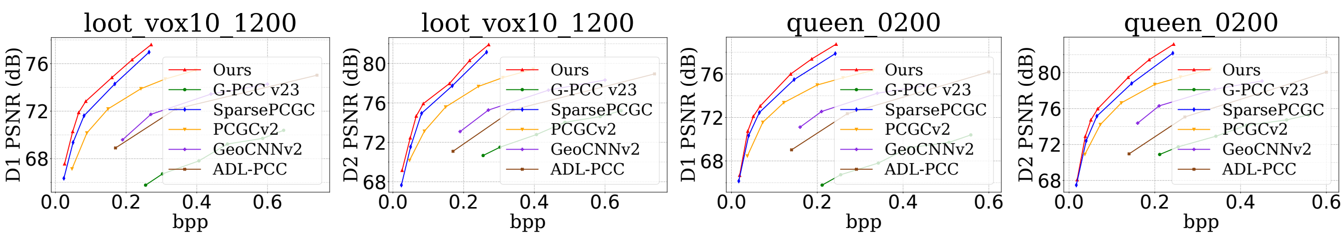

Figure 5: Performance comparison using rate-distortion curves.

where $T \in \{\lambda_0, \lambda_1 \cdots \lambda_n\}$, T is the preset modulation factor set and we calculate the total loss to train models of different complexity and rate.

Experimental Results

Experiment Setup

Dataset. In our experiment, we use the ShapeNet dataset and the selected dense point clouds. ShapeNet (Chang et al. 2015) is a 3D object CAD model dataset, encompassing a subset known as ShapeNet-Core. The subset comprises 55 common object categories and approximately 51,300 unique 3D mesh models. We densely sample points on the original meshes to generate point clouds. Subsequently, we randomly rotate and quantize them with 8-bit geometry precision in each dimension. We use the processed shapenet dataset for training of the proposed UniPCGC. For dense point clouds, we choose from MPEG PCC dataset, including longdress 1300, redandblack 1550, soldier 0690, loot 1200, queen 0200, basketball player 0200, and dancer 0001. We select these point cloud as testset for lossy and lossless compression. We also employ the 8iVFB and MVUB sequences for training and 8iVFB sequences for testing to align with popular lossless compression methods.

Training Strategies. We use MinkowskiEngine (Choy, Gwak, and Savarese 2019) and Pytorch to build our model, and perform UELC and VRCM training on RTX4080 GPU and intel i5-13600KF CPU. For UELC, it is trained for lossless compression using the loss function equation (7). We initialize the learning rate to 8×10^{-4} and gradually decrease it to 5×10^{-5} during training. We use the Adam optimizer and train for 30 epochs with a batch size of 8. For VRCM, it is trained for lossy compression using the loss function equation (8) and (9). We initialize the learning rate to 8×10^{-4} and gradually decrease it to 1×10^{-5} during training. We use the Adam optimizer and train for 20 epochs with a batch size of 8. We set the λ to 0.3 and 3.0.

Lossless Compression Results

Compared to G-PCC, we achieve a compression ratio (CR)-gain of 19.6%. Compared to SparsePCGC, we achieve a CR-gain of 5.6% when training dataset is ShapeNet, as shown in Supplementary Material.

To compare with Voxeldnn, Octattention and ECM-OPCC methods, we train UELC using 8iVFB and MVUB and the test results are shown in Figure 1 and Table 1. UELC demonstrates improved performance on the 8iVFB testset compared to SparsePCGC, with a 8.1% improvement. It also outperforms Voxeldnn (Nguyen et al. 2021a) with a 17.4% improvement, MsVoxelDNN (Nguyen et al. 2021b) with a 35.2% improvement, Octattention(Fu et al. 2022) with a 21.2% improvement, GPCC v23 with a significant 29.2% improvement and ECM-OPCC (Jin et al. 2024) with a 11.3% improvement. More detailed compression experiments are in the supplementary material. Compared to the previous SOTA methods, our model achieves a performance gain of 8.1%. Notably, our model exhibits a small model size (Gao et al. 2022) and fast inference speed, as shown in Figure 1. To evaluate the generalization of UELC, we also use UELC for lidar point cloud compression. More experimental results are shown in the Supplementary Material.

Lossy Compression Results

In the lossy compression, we set n as 0,1,2 to achieve variable rate within different ranges. Due to space limitations, table in the Supplementary Material presents the comparison between our approach and the latest traditional method G-PCC. Using D1 and D2 as distortion metrics, our method

Dense Point Clouds	PCGCv2		ADL-PCC		GeoCNNv2		SparsePCGC	
	D1	**D2**	**D1**	**D2**	**D1**	**D2**	**D1**	**D2**
longdress_vox10_1300	-49.76%	-41.73%	-77.90%	-76.34%	-75.17%	-70.19%	-14.15%	-14.47%
loot_vox10_1200	-50.31%	-42.85%	-78.48%	-76.86%	-76.43%	-71.27%	-19.95%	-14.85%
red&black_vox10_1550	-49.16%	-36.54%	-71.20%	-70.03%	-73.07%	-66.03%	-11.09%	-9.43%
soldier_vox10_0690	-48.57%	-41.20%	-74.28%	-72.46%	-73.84%	-68.45%	-15.91%	-12.63%
queen_vox10_0200	-41.30%	-37.77%	-79.07%	-78.64%	-73.27%	-67.80%	-12.15%	-12.79%
player_vox11_0200	-53.86%	-48.58%	-83.03%	-81.79%	-80.77%	-76.86%	-12.44%	-11.41%
dancer_vox11_0001	-53.00%	-45.32%	-81.56%	-78.39%	-79.01%	-72.40%	-12.43%	-9.80%
Average	**-49.75%**	**-42.00%**	**-77.93%**	**-76.36%**	**-75.94%**	**-70.43%**	**-14.02%**	**-12.20%**

Table 2: BD-Rate gains measured using both D1 and D2 for the UniPCGC against the PCGCv2, ADL-PCC, GeoCNNv2 and SparsePCGC for lossy compression.

Method	SG	NSG	UEG	Gain	Parms
Baseline	✓			0.0%	5.0M
Baseline+NSG		✓		-2.0%	1.3M
Baseline+UE	✓		✓	-1.8%	2.3M
UELC		✓	✓	**-5.6%**	2.3M

Table 3: The ablation experiments of UELC.

Method	UELC	VR	VC	D1-Gain	D2-Gain
Base				0.0%	0.0%
UELC	✓			-2.03%	-2.09%
VRM	✓	✓		-8.20%	-7.07%
VRCM	✓	✓	✓	**-14.02%**	**-12.20%**

Table 4: The ablation experiments of VRCM.

shows an average Bjontegaard Delta (BD)-rate improvement of 93.26% and 88.47% over the latest G-PCC v23. We also demonstrate superior performance compared to learning-based methods. VRCM outperforms SparsePCGC (Wang et al. 2022) by 14.02% and 12.20% in BD-rate, PCGCv2 (Wang et al. 2021) by 49.75% and 42.00% in BD-rate, GeoCNNv2 (Quach, Valenzise, and Dufaux 2020) by 75.94% and 70.43% in BD-rate, and ADL-PCC (Pereira 2021) by 77.93% and 76.36% in BD-rate, as shown in Tabel 2 and Figure 5. We not only achieve state-of-the-art RD performance but also support variable rate and complexity.

By setting different values for λ and n, fine-grained rate modulation can be achieved. When $\lambda \in (-20, 3)$ and $n = 0, 1, 2$, continuous rate variation can be achieved. Due to GPU memory limitations (less than 16GB), we use only two values of λ during model training, which already encompass a wide range of rates. If more values of λ are used during model training, the RD performance is expected to improve further. Moreover, our model achieves faster inference speed at low rate. Compared to static models, our dynamic network model achieves a encoding acceleration of 59.6% at the lowest rate and a decoding acceleration of 33.4% in loot 1200. We provide RD curves for variable rate and complexity in the Supplementary Material.

Ablation Studies

Table 3 presents the ablation results of proposed UELC. In UELC, we propose two techniques: Uneven Eight grouping (UEG) and Non-Sequential grouping (NSG). SparsePCGC is used as the baseline, which employs sequential grouping (SG) and uniform 8 grouping scheme. In UEG, a larger distance is considered within each group, so we set k to 5 for feature extraction. To balance the cost introduced by $k = 5$, we set c to 16 as a trade-off between performance and complexity. UELC combines NSG and UEG, resulting in a parameter count of 2.3M and achieving a 5.6% CR-gain compared to SparsePCGC, which demonstrates the effectiveness of the UEG and NSG schemes. Table 4 is the ablation experiment of VRCM. D1/D2-Gain represents the BD(D1/D2)-rate gain over the baseline. VR stands for the proposed rate modulation module, and VC stands for the proposed dynamic sparse convolution. Combining the baseline with UELC results in BD-rate gains of 2.03% and 2.09%. Building upon this, the introduction of the variable rate control module not only brings about BD-rate gains of 8.20% and 7.07%, but also enables the model to have the capability of variable rate encoding. On the basis of UELC and VR, further gains of 14.02% and 12.20% in BD-rate are achieved by introducing dynamic sparse convolution, which proves the effectiveness of the proposed module.

Conclusion

The paper presents an efficient unified framework named UniPCGC, which supports lossless compression, lossy compression, variable rate and variable complexity. First, we introduce the UELC in the lossless mode, which allocates more computational complexity to groups with higher coding difficulty, and merges groups with lower coding difficulty. Second, VRCM is achieved in the lossy mode through joint adoption of a rate modulation module and dynamic sparse convolution. Compared with the previous state-of-the-art works, the proposed UniPCGC achieves a CR-gain of 8.1% on lossless compression, a BD-Rate gain of 14.02% on lossy compression, and supports variable rate and complexity. These advancement promote the flexibility and practicality of point cloud geometry compression.

Acknowledgments

This work was supported by The Major Key Project of PCL (PCL2024A02), Natural Science Foundation of China (62271013, 62031013), Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (2024B1212010006), Guangdong Province Pearl River Talent Program (2021QN020708), Guangdong Basic and Applied Basic Research Foundation (2024A1515010155), Shenzhen Science and Technology Program (JCYJ20240813160202004, JCYJ20230807120808017).

References

Ali, M. S.; Kim, Y.; Qamar, M.; Lim, S.-C.; Kim, D.; Zhang, C.; Bae, S.-H.; and Kim, H. Y. 2024. Towards efficient image compression without autoregressive models. *Advances in Neural Information Processing Systems*, 36.

Ballé, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*.

Biswas, S.; Liu, J.; Wong, K.; Wang, S.; and Urtasun, R. 2020. Muscle: Multi sweep compression of lidar using deep entropy models. *Advances in Neural Information Processing Systems*, 33: 22170–22181.

Bolukbasi, T.; Wang, J.; Dekel, O.; and Saligrama, V. 2017. Adaptive neural networks for efficient inference. In *International Conference on Machine Learning*, 527–536. PMLR.

Cao, C.; Preda, M.; Zakharchenko, V.; Jang, E. S.; and Zaharia, T. 2021. Compression of Sparse and Dense Dynamic Point Clouds—Methods and Standards. *Proceedings of the IEEE*, 109(9): 1537–1558.

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. 2015. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Choi, Y.; El-Khamy, M.; and Lee, J. 2019. Variable rate deep image compression with a conditional autoencoder. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3146–3154.

Choy, C.; Gwak, J.; and Savarese, S. 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3075–3084.

Cui, Z.; Wang, J.; Gao, S.; Guo, T.; Feng, Y.; and Bai, B. 2021. Asymmetric gained deep image compression with continuous rate adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10532–10541.

Fu, C.; Li, G.; Song, R.; Gao, W.; and Liu, S. 2022. Octattention: Octree-based large-scale contexts model for point cloud compression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 625–633.

Gao, W.; Ye, H.; Li, G.; Zheng, H.; Wu, Y.; and Xie, L. 2022. OpenPointCloud: An open-source algorithm library of deep learning based point cloud compression. In *Proceedings of the 30th ACM international conference on multimedia*, 7347–7350.

Graves, A. 2016. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*.

He, D.; Zheng, Y.; Sun, B.; Wang, Y.; and Qin, H. 2021. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14771–14780.

Huang, G.; Chen, D.; Li, T.; Wu, F.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*.

Huang, G.; Liu, S.; Van der Maaten, L.; and Weinberger, K. Q. 2018. Condensenet: An efficient densenet using learned group convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2752–2761.

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

Jia, C.; Ge, Z.; Wang, S.; Ma, S.; and Gao, W. 2022. Rate distortion characteristic modeling for neural image compression. In *2022 Data Compression Conference (DCC)*, 202–211. IEEE.

Jin, Y.; Zhu, Z.; Xu, T.; Lin, Y.; and Wang, Y. 2024. ECM-OPCC: Efficient Context Model for Octree-Based Point Cloud Compression. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7985–7989.

Lin, J.; Liu, D.; Liang, J.; Li, H.; and Wu, F. 2021. A deeply modulated scheme for variable-rate video compression. In *2021 IEEE International Conference on Image Processing (ICIP)*, 3722–3726. IEEE.

Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; and Zhang, C. 2017. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, 2736–2744.

Nguyen, D. T.; and Kaup, A. 2022. Learning-based lossless point cloud geometry coding using sparse tensors. In *2022 IEEE International Conference on Image Processing (ICIP)*, 2341–2345. IEEE.

Nguyen, D. T.; and Kaup, A. 2023. Lossless point cloud geometry and attribute compression using a learned conditional probability model. *IEEE Transactions on Circuits and Systems for Video Technology*.

Nguyen, D. T.; Quach, M.; Valenzise, G.; and Duhamel, P. 2021a. Learning-based lossless compression of 3d point cloud geometry. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4220–4224. IEEE.

Nguyen, D. T.; Quach, M.; Valenzise, G.; and Duhamel, P. 2021b. Multiscale deep context modeling for lossless point cloud geometry compression. *arXiv preprint arXiv:2104.09859*.

Pang, J.; Lodhi, M. A.; and Tian, D. 2022. GRASP-Net: Geometric residual analysis and synthesis for point cloud compression. In *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*, 11–19.

Pereira, A. G. N. R. F. 2021. Adaptive Deep Learning-Based Point Cloud Geometry Coding. *IEEE Journal on Selected Topics in Signal Processing*, 15: 415–430.

Quach, M.; Valenzise, G.; and Dufaux, F. 2020. Improved Deep Point Cloud Geometry Compression. *2020 IEEE MMSP Workshop*.

Song, M.; Choi, J.; and Han, B. 2021. Variable-rate deep image compression through spatially-adaptive feature transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2380–2389.

Song, R.; Fu, C.; Liu, S.; and Li, G. 2023. Efficient Hierarchical Entropy Model for Learned Point Cloud Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14368–14377.

Sun, Z.; Tan, Z.; Sun, X.; Zhang, F.; Qian, Y.; Li, D.; and Li, H. 2021. Interpolation variable rate image compression. In *Proceedings of the 29th ACM international conference on multimedia*, 5574–5582.

Tao, L.; and Gao, W. 2021. Efficient Channel Pruning Based on Architecture Alignment and Probability Model Bypassing. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 3232–3237.

Tao, L.; Gao, W.; Li, G.; and Zhang, C. 2023. Adanic: Towards practical neural image compression via dynamic transform routing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16879–16888.

Teerapittayanon, S.; McDanel, B.; and Kung, H.-T. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*, 2464–2469. IEEE.

Theis, L.; Shi, W.; Cunningham, A.; and Huszár, F. 2016. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations*.

Veit, A.; and Belongie, S. 2018. Convolutional networks with adaptive inference graphs. In *Proceedings of the European conference on computer vision (ECCV)*, 3–18.

Wang, J.; Ding, D.; Li, Z.; Feng, X.; Cao, C.; and Ma, Z. 2022. Sparse tensor-based multiscale representation for point cloud geometry compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, J.; Ding, D.; Li, Z.; and Ma, Z. 2021. Multiscale point cloud geometry compression. In *2021 Data Compression Conference (DCC)*, 73–82. IEEE.

Wang, J.; Zhu, H.; Ma, Z.; Chen, T.; Liu, H.; and Shen, Q. 2019. Learned point cloud geometry compression. *arXiv preprint arXiv:1909.12037*.

Wang, X.; Yu, F.; Dou, Z.-Y.; Darrell, T.; and Gonzalez, J. E. 2018. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European conference on computer vision (ECCV)*, 409–424.

Xie, L.; Gao, W.; Zheng, H.; and Li, G. 2024. ROI-Guided Point Cloud Geometry Compression Towards Human and Machine Vision. In *Proceedings of the 32nd ACM International Conference on Multimedia*.

Yang, F.; Herranz, L.; Van De Weijer, J.; Guitián, J. A. I.; López, A. M.; and Mozerov, M. G. 2020. Variable rate deep image compression with modulated autoencoder. *IEEE Signal Processing Letters*, 27: 331–335.

Yin, S.; Li, C.; Bao, Y.; Liang, Y.; Meng, F.; and Liu, W. 2022. Universal efficient variable-rate neural image compression. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025–2029. IEEE.

You, K.; Liu, K.; Yu, L.; Gao, P.; and Ding, D. 2024. Pointsoup: High-Performance and Extremely Low-Decoding-Latency Learned Geometry Codec for Large-Scale Point Cloud Scenes. In Larson, K., ed., *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, 5380–5388. International Joint Conferences on Artificial Intelligence Organization. Main Track.