APSIPA Transactions on Signal and Information Processing, 2025, 14, e106

Original Paper

Efficient Multi-stage Context Based Entropy Model for Learned Lossy Point Cloud Attribute Compression

Kai Wang[1], Pingping Zhang[2], Shengjie Jiao[1], Hui Yuan[3], Shiqi Wang[2] and Xu Wang[1]*

[1] *College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.*
[2] *Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong.*
[3] *School of Control Science and Engineering, Shandong University, Jinan, China.*

ABSTRACT

The autoregressive entropy model facilitates high compression efficiency by capturing intricate dependencies but suffers from slow decoding due to its serial context dependencies. To address this, we propose ParaPCAC, a lossy Parallel Point Cloud Attribute Compression scheme, designed to optimize the efficiency of the autoregressive entropy model. Our approach focuses on two main components: a parallel decoding strategy and a multi-stage context-based entropy model. In the parallel decoding strategy, we partition the voxels of the quantized latent features into non-overlapping groups for independent context entropy modeling, enabling parallel processing. The multi-stage context based entropy model is employed to decode neighboring features concurrently, utilizing previously decoded features at each stage. Global hyperprior is incorporated after the first stage to improve the estimation of attribute probability. Through these two techniques, ParaPCAC achieves

*Corresponding author: Xu Wang, wangxu@szu.edu.cn.

Received 30 July 2024; revised 21 November 2024; accepted 29 November 2024
ISSN 2048-7703; DOI 10.1561/116.20240051

significant decoding speed enhancements, with an acceleration of up to 160× and a 24.15% BD-Rate reduction compared to serial autoregressive entropy models. Furthermore, experimental results demonstrate that ParaPCAC outperforms existing learning-based methods in rate-distortion performance and decoding latency.

Keywords: Point cloud compression, point cloud attribute compression, learned data compression

1 Introduction

Point clouds, a prevalent format for representing 3D scene data, find extensive application in graphics and autonomous driving, among other fields. Comprising discrete sampled points on object surfaces, each point is defined by its geometry (expressed as [x, y, z] coordinates in 3D space) and attributes (including RGB color, surface normals, reflectance intensity, etc.). With large-scale point clouds often containing millions of points, there is a pressing demand for efficient compression techniques to manage their substantial data volumes. In recent years, learning-based point cloud compression has garnered significant interest, with extensive research conducted on techniques such as learning-based point cloud geometry compression [35, 32, 31].

However, exploration into learning-based point cloud attribute compression (PCAC) remains relatively limited. Two mainstream approaches to PCAC include hybrid coding frameworks and end-to-end frameworks. Hybrid codecs integrate learnable modules into traditional point cloud frameworks, e.g., MPEG standardized G-PCC, enhance compression performance [12, 10, 41, 28]. One well-known codec is 3DAC [12], which combines a Region-adaptive Hierarchical Transform (RAHT) [10] with a learning-based entropy model. GPCC++ [41] based on G-PCC employs learnable filters to mitigate distortion for decoded attributes [28]. While hybrid coding methods improve compression performance by combining traditional algorithms with learnable modules, the system complexity increases, making end-to-end optimization difficult. The integration and tuning costs of such frameworks are high, limiting their scalability and applicability in real-world applications. Thus, end-to-end PCAC has been proposed to facilitate comprehensive optimization [33, 34]. Wang *et al.* [34] introduced an entropy model that integrates joint hyperprior and autoregressive neighborhood context. Nonetheless, this serial process requires decoding each point or block step by step, significantly slowing down the decoding speed, especially when handling large-scale point cloud data, where the decoding bottleneck becomes highly evident. Besides, Wang *et al.* [33]

proposed a lossless PCAC model leveraging multiscale structures and cross-scale/group/color correlations for accurate probability estimation, but they lack consideration of the global dependency, which can enhance probability estimation.

To overcome the inefficiency of serial autoregressive decoding, we introduce a novel parallel decoding strategy. By partitioning the quantized latent features into several non-overlapping groups, our model enables independent context entropy modeling for each group, facilitating both inter-group and intra-group parallelism. This approach significantly accelerates the decoding process by allowing multiple parts of the data to be processed simultaneously.This parallel decoding approach ensures high efficiency without sacrificing compression performance. The global hyperprior is derived via an attention module, integrating global attribute latent and geometric features. To streamline the computational complexity of the global hyperprior, we choose to compress and transmit it directly to the decoder. In the multi-stage context based entropy decoding, the current features are decoded with the aid of the global hyperprior and previously decoded local features from the same group. Here, the global hyperprior functions as contextual information, enriching the understanding of the broader context.

- We present an efficient parallel decoding strategy tailored for lossy point cloud attribute compression, effectively boosting decoding efficiency without compromising decoding quality. Our model demonstrates remarkable speed enhancements, achieving up to a 160× acceleration and a 24.15% BD-Rate reduction compared to serial autoregressive entropy models.
- We propose an efficient multi-stage context based entropy model to capture both short-range and long-range dependencies from previously decoded features and global prior to enhance compression performance.
- Experimental results demonstrate that our method outperforms other end-to-end learned approaches in compression performance with an applicable decoding latency, as illustrated in Figure 1.

2 Related Works

2.1 Point Cloud Attribute Compression (PCAC)

Point cloud attribute compression methods can be broadly categorized into rule-based and learning-based methods. Rule-based methods typically rely on conventionally point cloud compression codec, while learning-based approaches tend to use neural networks to predict the probability distribution

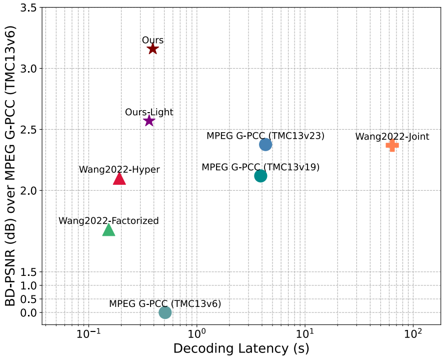

Figure 1: Comparison of average compression performance and decoding latency among various point cloud attribute compression codecs on the *longdress* (10-bit geometry) from 8iVFBv2 [8] dataset. Our proposed model, compared with the joint hyper-autoregressive entropy model [34], demonstrates notable decoding speed enhancements while maintaining comparable decoding latency to codecs utilizing hyperprior entropy model (Hyper) and factorized entropy models (Factorized). Notably, Ours-Light shares the same backbone architecture as Hyper and Factorized, but these models employ different entropy models.

of point cloud attributes. In this subsection, we provide a brief review of existing research from both these perspectives.

Rules based Methods typically rely on conventionally designed transformations, *e.g.*, Graph Fourier Transform (GFT) [39], RAHT [10], Gaussian Process Transform (GPTs) [11], and Region Adaptive Graph Fourier Transform (RA-GFT) [20], to leverage spatial correlations among the attribute values of points. GFT [39] and its variants [7, 29, 36] perform Laplace decomposition on the local graph of points and decompose the attributes by eigen decomposition. RAHT [10] performs regionally adaptive weighted multi-resolution wavelet transform on attributes in 3D space, which has competitive speed and coding performance. GPT [11] models the statistics of the attributes using stationary Gaussian processes (GPs) and applies a Karhunen-Loève transform for coding. RA-GFT [20] is a generalized version of RAFT that uses Q-normalized graph Laplacian. In the G-PCC [28] (TMC13), RAHT is incorporated as a core transformation due to its efficiency. Simultaneously, the TMC13 introduces rules-based entropy model and coefficient prediction to achieve efficient compression performance.

Learning based Methods have attracted attention in point cloud geometry [25, 35, 4, 26, 32, 31] and point cloud attribute [12, 34, 33] compression. These approaches tend to use neural networks to predict the probability distribution of point cloud attributes.

According to the learning strategy of the modules in codec, existing methods can be divided into hybrid approaches and end-to-end learning based approaches. The hybrid approaches replace parts (transformation or entropy model) of the traditional codec with learnable modules, or enhance traditional codec as a plugin. For lossless point cloud attribute compression, Wang *et al.* [33] and Nguyen *et al.* [19] proposed learnable entropy models by using sparse convolutional networks to estimate attribute distribution. For lossy compression, Quach *et al.* [24] attempted to fold 3D point clouds onto a 2D plane and directly compress them using learned image/video codecs. Fang *et al.* [12] proposed 3DAC using RAHT [10] as transformation and constructing a learnable entropy model to predict the probability distribution of high-frequency coefficients of attributes. YOGA [40] feeds the output of learnable transformation into G-PCC, while GPCC++ [41] filters the reconstructed result of G-PCC to remove distortion, achieving performance improvement compared to G-PCC.

End-to-end learning based approaches are believed to have a higher theoretical performance [3] as they enable the simultaneous optimization of the transformation and entropy model. Recently, Sheng *et al.* [30] used a point-based neural network for attribute compression, Pinheiro *et al.* [22] use a normalizing flow based network for attribute compression, and Wang *et al.* [34] utilized sparse convolutional networks [6] with introducing joint hyperpriors [3] and autoregressive contexts [18] based entropy model to improve attribute compression performance. However, end-to-end learning-based methods have not been able to achieve compression performance surpassing that of traditional point cloud codecs like G-PCC [1, 30], or they may exhibit unacceptable decoding latency [34].

2.2 Efficient Learning Based Entropy Models

Accurate entropy models are essential for improving compression performance. In image compression, joint hyperprior and autoregressive context entropy models [18] have been highly effective in reducing spatial redundancy and improving efficiency. These models typically combine a global hyperprior with local context models to predict the distribution of image representations. However, their sequential decoding process leads to slower speeds due to the autoregressive structure.

To address this, researchers have explored parallelization strategies. For instance, He [14] introduced a checkerboard context entropy model for image compression, dividing images into groups for partial parallel decoding. While this improves speed, it struggles to capture long-range dependencies due to the limited receptive field of CNNs. Kim [16] and Qian [23] incorporated global references into entropy models, which helped capture both local and global contexts, but at the cost of increased computational complexity.

Building on these advancements, similar entropy models [34] have been applied to point cloud attribute compression, exploiting local correlations to improve compression. Like in image compression, these models also face slow decoding due to their sequential nature. Wang [34] proposed a group-based model for partial parallel decoding, improving speed while maintaining compression quality.

However, both image and point cloud compression methods still struggle to capture long-range dependencies, particularly in 3D point clouds. Recent works have introduced attention mechanisms to overcome these limitations. For example, OctAttention [13] applied global attention to expand the receptive field, while Song [31] proposed EHEM, a hierarchical attention structure to improve efficiency and performance.

Inspired by these efforts, we propose a multi-stage context-based global entropy model for 3D point cloud attribute compression. By capturing both local and global dependencies through context-based and attention mechanisms, our model enables parallel decoding and improves compression performance. This approach strikes a balance between efficiency and complexity, making it well-suited for real-world 3D data compression applications.

3 Preliminary

3.1 Variational Point Cloud Attribute Compression

The VAE (Variational Autoencoder) based point cloud attribute compression framework consists of transformation and entropy coding. Given a voxelized point cloud $P = (\boldsymbol{G}, \boldsymbol{X})$ with known coordinates $\boldsymbol{G} \in \mathbb{Z}^{N\times 3}$, attribute compression only considers the representation and compression of attributes $\boldsymbol{X} \in [0,1]^{N\times S}$ where S is the number of channels and N is the number of non-empty voxels in sparse tensor.

In the first step of the encoding process, point cloud attributes $\boldsymbol{X}$ are transformed into quantized latent representations $\hat{\boldsymbol{Y}}$ through a learned analysis transform g_a and a scalar quantizer Q. In learning-based point cloud compression methods, the analysis transform g_a is typically employed to map the input data into a more compact latent space representation. This process involves a series of downsampling operations or feature extraction layers, which reduce the spatial and attribute redundancy, capturing the essential information required for efficient compression.

After transformation, the entropy model estimates the probability distribution $p_{\hat{\boldsymbol{Y}}}$ of the quantized latent representations $\hat{\boldsymbol{Y}}$, enabling efficient encoding into a bitstream. The goal of the compression process is to minimize the bit rate (denoted as $\boldsymbol{R}$), which represents the number of bits required to encode the latent variables. The bit rate $\boldsymbol{R}$ is defined as the expected negative log-likelihood of the estimated probability distribution, i.e., $\boldsymbol{R} = -\log_2 p_{\hat{\boldsymbol{Y}}}(\hat{\boldsymbol{Y}})$, and minimizing $\boldsymbol{R}$ ensures efficient encoding.

On the decoder side, the synthesis transform g_s reconstructs the attributes from the decoded latent representations $\hat{\boldsymbol{Y}}$. The synthesis transform g_s is the inverse process of g_a, aiming to recover the original point cloud attributes from the compact latent space representation. It performs a series of upsampling operations or feature decoding layers that map the latent variables back into the original high-dimensional space, restoring the detailed point cloud attributes. In essence, g_s functions as a decoder network that reverses the compression process, attempting to faithfully reconstruct the input data from its compressed latent form. The distortion (denoted as $\boldsymbol{D}$) measures the discrepancy between the original attributes $\boldsymbol{X}$ and the reconstructed attributes $\hat{\boldsymbol{X}} = g_s(\hat{\boldsymbol{Y}})$ after decoding. $\boldsymbol{D}$ is typically evaluated using a distortion metric such as the Binary Cross-Entropy (BCE) loss, quantifying how well the reconstructed attributes preserve the original information.

During the training phase, the objective is to minimize the combined rate-distortion loss function $\boldsymbol{L}$, which balances the trade-off between the bit rate and distortion. The loss function is defined as follows:

$$\begin{aligned}\mathcal{L} &= R + \lambda \cdot D \\ &= \mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}}}[-\log_2 p_{\hat{\boldsymbol{Y}}}(\hat{\boldsymbol{Y}})] + \lambda \cdot \mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}}}[d(\boldsymbol{X}, g_s(\hat{\boldsymbol{Y}}))],\end{aligned} \tag{1}$$

where $p_{\boldsymbol{X}}$ is the real distribution of the point cloud attributes, d is the distortion metric, and λ is the Lagrangian multiplier used to control the rate-distortion trade-off.

3.2 Joint Hyper-Autoregressive Entropy Model

The accuracy of estimating the distribution p is crucial for entropy coding. Various entropy models have been proposed and used in 2D image compression, such as factorized, hyper [3], and autoregressive prior entropy models and their combinations [18]. For 3D point cloud attributes, Wang *et al.* [34] first explored estimating the distribution $p_{\hat{\boldsymbol{Y}}}$ using a joint of hyperprior and autoregressive context. With the joint entropy model, the rate R term in the rate-distortion trade-off loss function can be written as:

$$R = \mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}}}[-\log_2 p_{\hat{\boldsymbol{Y}}|\hat{\boldsymbol{Z}}_l}(\hat{\boldsymbol{Y}} \mid \hat{\boldsymbol{Z}}_l) - \log_2 p_{\hat{\boldsymbol{Z}}_l}(\hat{\boldsymbol{Z}}_l)], \tag{2}$$

where $\hat{\boldsymbol{Z}}_l = Q(h_a(\boldsymbol{Y}))$ is the hyperprior used to eliminate neighborhood redundancy and $p_{\hat{\boldsymbol{Z}}_l}$ is the estimated distribution of the hyperprior.

The probability estimation of the joint entropy model can be modeled as:

$$\begin{gathered} p_{\hat{\boldsymbol{Y}}|\hat{\boldsymbol{Z}}_l}(\hat{\boldsymbol{Y}} \mid \hat{\boldsymbol{Z}}_l) = \prod_i (\mathcal{V}(\mu_i, \sigma_i) * \mathcal{U}(-\frac{1}{2}, \frac{1}{2}))(\hat{Y}_i), \\ \text{with } \mu_i, \sigma_i = g_{ep}(g_{cm}(\hat{\boldsymbol{Y}}_{<i}), \boldsymbol{\psi}), \boldsymbol{\psi} = h_s(\hat{\boldsymbol{Z}}_l). \end{gathered} \tag{3}$$

$\mathcal{V}$ represents a probability distribution, *e.g.*, Laplace distribution and Gaussian distribution. $\mathcal{U}$ is an uniform distribution. g_{ep} and g_{cm} are the entropy model and the autoregressive context model, respectively. And, h_s is the hyperpripor decoder. For decoding the i-th quantized latent $\hat{Y}_i$ in a point cloud with morton order [2], the decoded latents $\hat{\boldsymbol{Y}}_{<i}$ are used to generate the autoregressive context by the autoregressive context model g_{cm} implemented by a masked sparse convolution. The autoregressive context is then concatenated with the output of the hyperprior decoder $h_s(\hat{\boldsymbol{Z}}_l)$ to generate the parameters (μ_i, σ_i) of probability distribution, *e.g.*, mean / diversity for the Laplace distribution or mean / standard deviation for the Gaussian distribution.

Unlike images defined on 2D dense grids, point cloud attributes exist on the non-empty voxels of the determined 3D sparse grids. Due to the sparsity and unorder of the input point cloud, it is hard to effectively capture the neighbor context of the current non-empty voxel to be encoded, which limits the performance improvement brought by the autoregressive context model. Additionally, autoregressive context model has an unacceptable decoding latency due to serial context dependencies.

4 Proposed Method

The overview of the proposed compression architecture is shown in Figure 2. The VAE based point cloud attribute compression architecture is employed as the backbone. Initially, the transformation network transforms the input point cloud attributes into latent features. Subsequently, these features are encoded into a bitstream by the entropy model. To enhance the decoding speed, we propose a parallel decoding strategy(Section 4.1) that does not rely on intra-group autoregressive context. The latent features are divided into K non-overlapping groups according to their coordinates to encode and decode them group-by-group. To compensate the loss of autoregressive context prior, we introduce a inter-group global context (Section 4.2). This approach leverages global context from previously decoded groups to provide context information for the groups to be decoded later. The proposed multi-stage context based entropy model (Section 4.3) integrates these methods, forming a comprehensive solution that optimizes both speed and compression performance.

4.1 Parallel Decoding Strategy

To address the problem of the slow decoding speed caused by the serial dependency of the autoregressive context model, we propose a multi-stage context modeling to support parallel decoding. Specifically, the quantized latent, generated via the 3D sparse convolution, contains many voxels, which then are

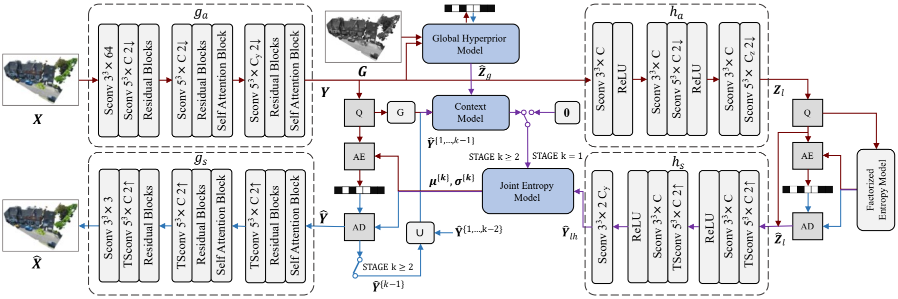
g_a
Sconv $3^3\times 64$
Sconv $5^3\times C\ 2\downarrow$
Residual Blocks
Sconv $5^3\times C\ 2\downarrow$
Residual Blocks
Self Attention Block
Sconv $5^3\times C_y\ 2\downarrow$
Residual Blocks
Self Attention Block
$\boldsymbol{X}$
$\boldsymbol{Y}$
$\boldsymbol{G}$
Global Hyperprior Model
$\widehat{\boldsymbol{Z}}_g$
Q
G
Context Model
$\boldsymbol{0}$
AE
$\widehat{\boldsymbol{Y}}^{\{1,\dots,k-1\}}$
STAGE k ≥ 2
STAGE k = 1
$\boldsymbol{\mu}^{\{k\}}, \boldsymbol{\sigma}^{\{k\}}$
Joint Entropy Model
AD
U
$\widehat{\boldsymbol{Y}}^{\{1,\dots,k-2\}}$
STAGE k ≥ 2
$\widehat{\boldsymbol{Y}}^{\{k-1\}}$
$\widehat{\boldsymbol{Y}}$
g_s
Sconv $3^3\times 3$
TSconv $5^3\times C\ 2\uparrow$
Residual Blocks
TSconv $5^3\times C\ 2\uparrow$
Residual Blocks
Self Attention Block
TSconv $5^3\times C\ 2\uparrow$
Residual Blocks
Self Attention Block
$\widehat{\boldsymbol{X}}$
h_a
Sconv $3^3\times C$
ReLU
Sconv $3^3\times C$
Sconv $5^3\times C\ 2\downarrow$
ReLU
Sconv $3^3\times C$
Sconv $5^3\times C_z\ 2\downarrow$
$\boldsymbol{Z}_l$
Q
AE
AD
Factorized Entropy Model
$\widehat{\boldsymbol{Z}}_l$
h_s
Sconv $3^3\times 2\ C_y$
ReLU
Sconv $3^3\times C$
TSconv $5^3\times C\ 2\uparrow$
ReLU
Sconv $3^3\times C$
TSconv $5^3\times C\ 2\uparrow$
$\widehat{\boldsymbol{Y}}_{lh}$

Figure 2: The overview of our proposed method. The left side shows the attributes autoencoder, and the right side shows the entropy model. "SConv $n^3 \times C$" and "TSConv $n^3 \times C$" denotes the sparse convolution and transposed convolution with C output channels and kernel size n^3. "Residual Block" and "Self Attention Block" represent the residual network and the local self attention network used for efficient latent feature aggregation. "$s\uparrow$" and "$s\downarrow$" represent upsampling and downsampling at a factor of s. "Q" represents quantizer, "AE" represent arithmetic encoder, and "AD" represent arithmetic decoder. "G" represents the partition operation of the quantized latent representations. "$\mathbf{0}$" symbolizes the context with the same shape as the input point cloud, where all attribute values are $\mathbf{0}$. $\cup$ is used to describe the combination of point clouds of different shapes. Red arrows represent the encoding data flow, blue arrows represent the decoding data flow, and purple arrows represent the shared data flow.

partitioned into several non-overlapped groups. For convenient description, we define a cube with the size of $2 \times 2 \times 2$, a total of 8 voxels, as a basic coding unit. These eight voxels in one cube are then partitioned into K groups according to their indexes in the cube. Consequently, the original encoding/decoding process becomes K-pass. During the implementation, after all the non-empty voxels in group k are encoded (*or decoded*), the encoding (*or decoding*) of the next group can begin. The context model of each non-empty voxel within the same group depends on the information of the non-empty voxels in the previously encoded (*or decoded*) groups. Since there is no serial dependency within a group, and the decoding of each group depends on the previously decoded groups, the non-empty voxels within the same group can be parallel decoded.

To achieve a trade-off between the compression performance and computing efficiency, we partition the downsampled voxelized point cloud output by g_s into 3 groups based on their spatial coordinates as shown in Figure 3. The first group $\{1\}$ can be regarded as low-scale context, while the second group $\{3, 6, 8\}$ and the third group $\{2, 4, 5, 7\}$ are interlaced to maximize the available neighborhood context. Our proposed multi-stage context based context structure can enhance probability estimation accuracy by leveraging the spatial correlation between groups with high decoding speed compared with the serial autoregressive context model.

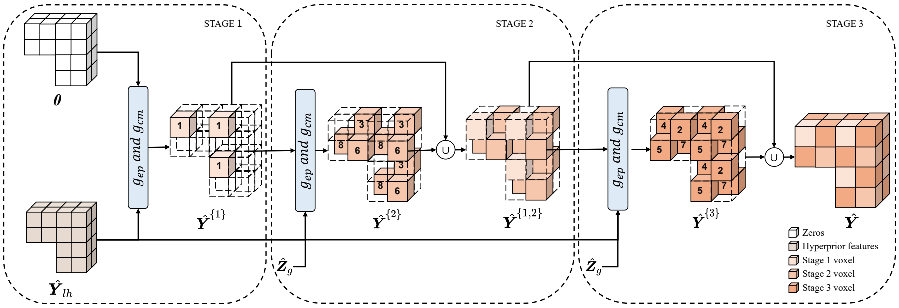

Figure 3: An example of a multi-stage context modeling scheme with three groups:{1}, {3,6,8}, {2,4,5,7}, aimed at enhancing the accuracy of probability estimation in an entropy model by exploiting spatial correlations among groups. **0** symbolizes the context with the same shape as the input point cloud, where all attribute values are 0. Y_{th} represents the hyperprior context output by decoder h_s, and $\hat{Z}_g$ defines the global hyperprior. The entropy decoding decodes the bitstream into voxels using the entropy model parameters$\{\mu, \theta\}$ generated by g_{ep} and g_{cm}.

4.2 Global Context Model

Despite the parallel decoding strategy improving decoding speed, it unfortunately results in the loss of context information. Moreover, the entropy model can only capture short-range information (*i.e.*, local context) due to the limited receptive field of sparse convolution networks. Consequently, the full utilization of long-range information (*i.e.*, global context) is hindered.

To address this issue, we introduce an inter-group global context, as illustrated in Figure 4. This global context is designed to enhance the efficiency of the inter-group context information utilization. It adaptively selects global context information from previously decoded groups using the attention mechanism. To avoid quadratic computational complexity of global attention, a global hyperprior $\boldsymbol{Z}_g$ is employed inspired by Informer [16]. The global hyperprior is generated as follows:

$$\boldsymbol{Z}_g = h_g(f_g(\boldsymbol{G}), \boldsymbol{Y}; \boldsymbol{\tau}), \tag{4}$$

where h_g is the global hyperprior encoder with a multi-head attention block, $f_g(\boldsymbol{G})$ is the point cloud geometry feature and $\boldsymbol{\tau}$ is a learnable parameter with a fixed length. In order to obtain the global hyperprior during the decoder stage, the global hyperprior $\boldsymbol{Z}_g$ is quantized to $\hat{\boldsymbol{Z}}_g$ and compressed to bitstream by a factorized entropy model [3] for transmission.

4.3 Multi-stage Context Based Entropy Model

The multi-stage context-based entropy model is designed to improve the accuracy of attribute probability estimation by integrating both short-range and

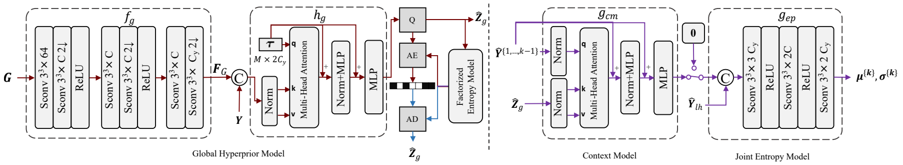

Figure 4: The overview of the Global Hyperprior Model and the proposed parallel Context Model and Joint Entropy Model.

long-range dependencies. This model operates by progressively refining the probability estimations using previously decoded features and global contextual information.

The workflow of the model is illustrated in Figure 3, which shows an example with three groups of voxel features: 1, 3, 6, 8, and 2, 4, 5, 7. These numbers represent the indices of voxel features in each group, and the features within the same group are encoded and decoded in parallel.

In the first stage, when no context is available, the model initializes the context to $\mathbf{0}$, a tensor of all zeros with the same shape as the expected output. During this stage, the hyperprior context $\hat{Y_{lh}}$ is used along with $\mathbf{0}$ to decode the first group of voxel features, denoted as 1. For subsequent stages, the entropy model is conditioned on both the local hyperprior $h_s(\hat{\boldsymbol{Z}}_l))$ and the global context $g_{cm}(\hat{\mathbf{Y}}^{\{1,\dots,k-1\}}, \hat{\boldsymbol{Z}}_g)$, which is derived from the previously decoded groups.

The parameters of the probability model for the k-th group are estimated as follows:

$$\mathbf{\Phi}^k = \begin{cases} g_{ep}(\mathbf{0}, h_s(\hat{\boldsymbol{Z}}_l)), & k = 1 \\ g_{ep}(g_{cm}(\hat{\mathbf{Y}}^{\{1,\dots,k-1\}}, \hat{\boldsymbol{Z}}_g), h_s(\hat{\boldsymbol{Z}}_l)), & k \geq 2 \end{cases} \tag{5}$$

where $\mathbf{\Phi}^k = \{\boldsymbol{\mu}^k, \boldsymbol{\sigma}^k\}$ represents the estimated parameters for the k-th group, and $\mathbf{0}$ is an all-zero tensor. The global context g_{cm} helps in refining the probability estimation for the current group by using information from the previously decoded groups.

After incorporating the global context, the rate term $\mathbf{R}$ in the rate-distortion loss function Eq. (1)is updated as follows:

$$\begin{aligned} R = \mathbb{E}_{\boldsymbol{X}\sim p_{\boldsymbol{X}}} [& -\log_2 p_{\hat{\mathbf{Y}}|\hat{\boldsymbol{Z}}_l,\hat{\boldsymbol{Z}}_g}(\hat{\mathbf{Y}} \mid \hat{\boldsymbol{Z}}_l, \hat{\boldsymbol{Z}}_g) \\ & - \log_2 p_{\hat{\boldsymbol{Z}}_l}(\hat{\boldsymbol{Z}}_l) - \log_2 p_{\hat{\boldsymbol{Z}}_g}(\hat{\boldsymbol{Z}}_g)], \end{aligned} \tag{6}$$

where $p_{\hat{\boldsymbol{Z}}_g}$ is the estimated distribution of the global hyperprior. This integration of global context enhances the models ability to capture both local and long-range dependencies, leading to more accurate probability estimations and improved compression performance.

5 Experimental Results

5.1 Datasets

We evaluate the proposed method on three datasets with different geometric and attribute characteristics.

Human body. The human body point cloud is widely used in the MPEG common test for evaluation. We select 8 point clouds include 4 point clouds from 8i Voxelized Full Bodies (8iVFBv2) [8] and 4 point clouds from Owlii Dynamic Human Textured Mesh Sequence Dataset (Owlii) [38] for evaluation. Each point cloud consists of approximately one million points.

Synthesize COCO. Due to the limited diversity in the geometry and color information of human body point clouds, we synthesize 10,000 colored point clouds for training using the COCO dataset [17]. To generate synthetic point clouds from the COCO dataset, we randomly select images from the dataset to use as the attributes for the synthetic point clouds. The geometry of these synthetic point clouds is based on a grid structure, which is then distorted using Perlin noise [21]. to introduce variability in the geometric shape. This noise simulates more natural and irregular surfaces within the point cloud. Furthermore, we apply random 3D rotations to the synthetic point clouds to further increase the diversity of the dataset. We train our model using the Synthesize COCO dataset and evaluated it on the Human Body dataset. Despite the differences in characteristics between the synthetic dataset and the human body dataset, the proposed method demonstrates strong generalization capability.

Indoor scene. The ScanNet dataset [9] is widely used for training and testing on indoor scenes, comprising 1603 scans of various indoor environments. We select 1503 scans for training and 100 scans for testing. Each scan contains a point cloud with approximately 0.8 million points.

Large-scale outdoor scene. The SensatUrban [15] dataset is a large-scale urban point cloud dataset collected by UAV photogrammetry. It contains about 6 billion points from two cities and covers about 6 square kilometers of the urban area. The point clouds are partitioned into $35m \times 35m$ patches according to the actual physical distance, each patch containing about 0.5 million points. We randomly select 1570 patches for training and 100 patches for testing.

5.2 Implementation Details

Training strategies. We implement the proposed methods using PyTorch and MinkowskiEngine [6]. Models are trained for 80 epochs on the synthetic dataset or 150 epochs on indoor and large-scale outdoor scene datasets using the Adam optimizer. We initiated training with an initial learning rate of

1×10^{-4}, which decayed by a factor of 0.3 every 30 epochs, with a lower bound of 2.7×10^{-6}. For training models at different bitrates, we set λ to the following values: 0.0006, 0.0018, 0.0130, 0.0063, and 0.0530.

Evaluation metrics. We utilize the Peak Signal-to-Noise Ratio (PSNR, in dB) of the Y channels at different bits per point (bpp) to evaluate the compression performance. To measure the averaged R-D performance of the proposed method, we also provide the Bjøntegaard Delta PSNR [5] (BD-PSNR, in dB) gain and the Bjøntegaard Delta bit rate [5] (BD-RATE, in percentage) reduction for comparison.

5.3 Performance Comparisons

Comparison to the G-PCC. We conduct the official implementation with three different versions of G-PCC (TMC13), including TMC13v6, TMC13v19 and TMC13v23. Following the MPEG common test conditions (CTC) with QPs={51, 46, 40, 34, 28}.

The rate-distortion (RD) performances are illustrated in Figure 5, and quantitative comparison results are shown in Table 1.

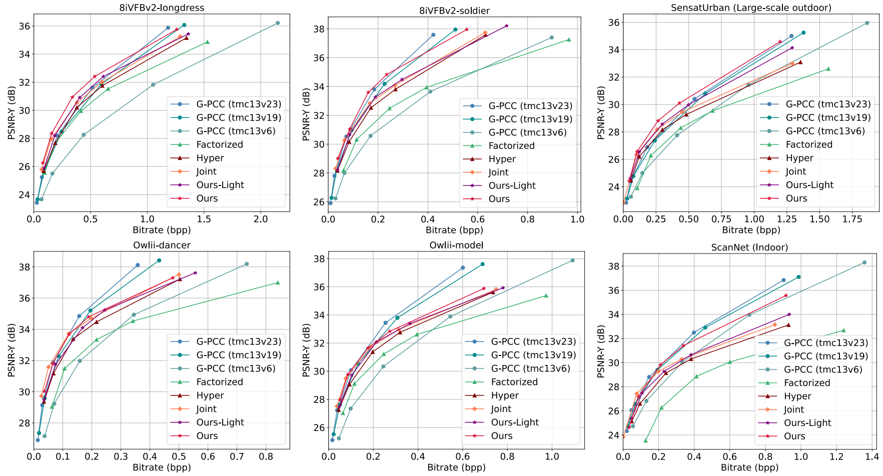

Figure 5: Rate-distortion curves of various point cloud attribute compression approaches. The results are evaluated on Human Body (8iVFBv2, Owlii), ScanNet and SensatUrban datasets.

Our model demonstrates a superior capability in handling point clouds with intricate textures. Specifically, our model outperform the latest version of MPEG G-PCC(TMC13v23) with 0.74dB and 0.05dB BD-PSNR improvement and 17.58% and 1.29% BD-Rate reductions on the longdress and soldier point clouds, respectively.

Table 1: Comparison results of the proposed method with the G-PCC and other learning based methods in terms of BD-PSNR(Y) (dB) and BD-RATE (%).

Dataset	Point Cloud	Ours vs G-PCC(TMC13v23)		Ours vs Hyper		Ours vs Joint	
		BD-PSNR (dB) ↑	BD-RATE (%) ↓	BD-PSNR (dB) ↑	BD-RATE (%) ↓	BD-PSNR (dB) ↑	BD-RATE (%) ↓
8iVFBv2	longdress	+0.74	–17.58	+1.09	-26.81	+0.57	-16.19
	loot	-1.09	+50.54	+0.60	-17.80	-0.16	+7.59
	redandblack	-0.66	+24.20	+0.55	-15.43	-0.19	+8.42
	soldier	+0.05	-1.29	+1.10	-28.66	+0.62	-18.98
	Average	-0.24	+13.95	+0.84	-22.18	+0.21	-4.79
Owlii	basketball_player	-0.23	+9.73	+0.67	-21.14	-0.08	+4.03
	dancer	-0.42	+14.71	+0.60	-18.92	-0.10	+4.82
	exercise	-0.55	+28.79	+0.35	-15.16	-0.27	+17.44
	model	-0.56	+17.01	+0.62	-18.20	+0.10	-2.75
	Average	-0.44	+17.56	+0.56	-18.36	-0.09	+5.89
ScanNet	**Average**	-0.44	+14.63	+1.25	-31.61	+0.25	+18.30
SensatUrban	**Average**	+0.70	-17.33	+1.06	-30.27	+0.52	-6.77

The visual quality comparison of the reconstructed results is presented in Figure 6 and Figure 7. G-PCC is a meticulously designed model renowned for its outstanding performance in compressing dense point clouds, particularly excelling in smooth scenarios such as human body point clouds (e.g., *loot*). As illustrated in Figure 6, our method achieves visual quality nearly indistinguishable from MPEG G-PCC (TMC13v23). Additionally, Figure 7 provides a detailed visualization of the *longdress* point cloud, highlighting the challenges G-PCC faces when dealing with point clouds containing complex textures. In contrast, our method effectively encodes point clouds with millions of points seamlessly, without requiring partitioning, thereby avoiding blocking artifacts. As a result, the reconstructed attributes generated by our approach demonstrate competitive perceptual quality, particularly in scenarios with intricate details.

Comparison to other learning based methods. We evaluate our proposed method by conducting a comprehensive comparison with other learning based point cloud attribute compression methods. These include the evaluating of performance against the baseline methods developed by Wang *et al.* [34], which employ the factorized entropy model referred to as "Factorized", the hyperprior entropy model referred to as "Hyper", and the joint autoregressive and hyperpriors entropy model referred to as "Joint", respectively. The R-D results shown in Figure 5 and the quantitative comparisons presented in Table 1 illustrate that our method outperforms "Hyper" across all datasets. We also present a visual comparison with the "Joint" model in the Figure 6 and Fig. 7. The results demonstrate that our method achieves more accurate color reproduction at comparable bitrates, particularly exhibiting superior clarity in regions where color blocks are adjacent.

In addition, we compared our method with the improved GPCC-based approach and non-standard entropy model-based approaches. The PDE-Based method [37], which is an improvement on GPCCv19, introduces a prediction

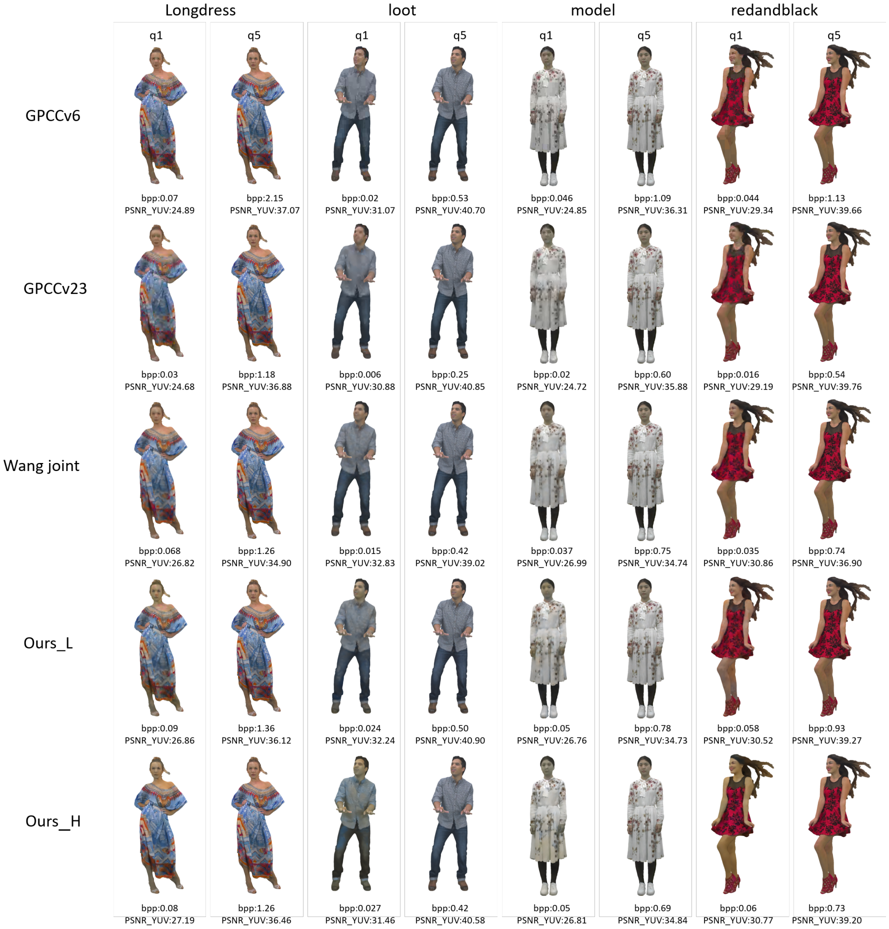

Figure 6: Visual quality comparison of the reconstructed point cloud on 8iVFBv2 dataset.

module based on partial differential equations (PDE), optimizing attribute gradients for prediction and fully utilizing the geometric distribution of adjacent regions to enhance point cloud attribute compression performance. The Progressive method [27], on the other hand, proposes a progressive coding model that gradually compresses the quantization residuals of the previous representation, allowing for a step-by-step improvement in the encoding of point cloud attributes. The R-D results in the Figure 8 show that our method outperforms both approaches across all datasets.

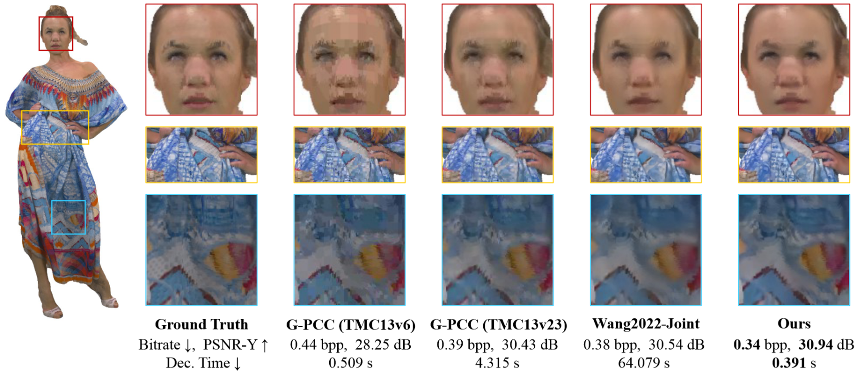

Figure 7: Visual quality detail comparison of the reconstructed point cloud on *Longdress* from the 8iVFBv2 dataset.

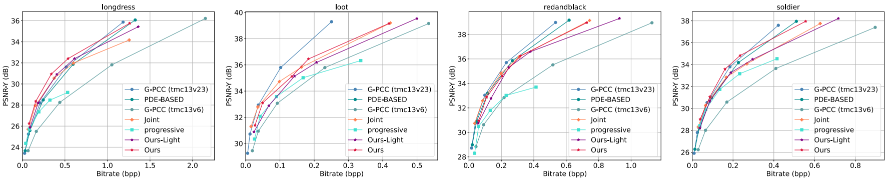

Figure 8: Rate distortion performance with other learning based methods.

5.4 Runtime Comparison

We compare the runtime of different methods in Table 2. For G-PCC, only the attribute encoding and decoding time are recorded to ensure a fair comparison. For learning based methods with GPU acceleration, we additionally record the GPU memory usage and the number of parameters of the network. The experiments are conducted utilizing an Intel Xeon Silver 4210R CPU and an NVIDIA Geforce RTX 3090 GPU. Our method achieves remarkable decoding speed enhancements, with an acceleration of up to 160×. For instance, it only takes 0.391 seconds to decode a point cloud containing 0.8 million points, whereas "Joint" requires around 64.079 seconds. Ours-Light shares the same backbone framework (Transform: L) as "Factorized" and "Hyper". Compared to Ours, Ours-Light reduces the encoding time by approximately 13%, and it also cuts the decoding time by about 7%, while only slightly sacrificing the BD-PSNR metric.

Table 2: Complexity and compression performance comparison among different methods on "longdress" point cloud (Anchor: G-PCC). The "L" denotes a lightweight transform network comprising of stacked convolutional layers, while "H" represents a heavyweight network incorporating self-attention layers.

Methods	Transform	#Param. ↓	GPU Mem. (GB) ↓	Enc. Time (s) ↓	Dec. Time (s) ↓	BD-PSNR (dB) ↑	BD-Rate (%) ↓
G-PCC	RAHT	-	-	0.700	0.509	-	-
G-PCC (TMC13v19)	RAHT	-	-	4.281	3.889	+2.12	-45.49
G-PCC (TMC13v23)	RAHT	-	-	5.062	4.315	+2.38	-49.75
Factorized	L	3.554M	1.903	0.148	0.153	+1.68	-37.22
Hyper	L	8.758M	1.911	0.210	0.192	+2.10	-44.33
Joint (Hyper + Autoregressive)	L	9.872M	1.848	0.189	64.079	+2.37	-53.17
Ours-Light w/o Global	L	9.872M	1.913	0.313	0.222	+2.17	-45.36
Ours-Light	L	18.103M	1.919	0.442	0.362	+2.57	-50.48
Ours	H	31.127M	2.491	0.513	0.391	+3.16	-57.90

5.5 Ablation Studies

We compare different entropy models (V1-V5), models with and without the multi-stage context (MSC) based entropy model (V3-V4), and models with and without the integration of global context (V4-V5). In addition, we conduct ablation studies on various transform networks (V5-V6), specifically, the lightweight (L) network utilizing only residual convolutional layers and the relatively heavyweight (H) transform network incorporating self-attention layers. Note that evaluations for V1-V5 are performed on the lightweight network.

The results are presented in Figure 9 and Table 3. Comparing the entropy models (V1 and V2) with the serial context based entropy model (V3), we observe that V3 outperforms in terms of BD-PSNR and BD-Rate, albeit with a longer decoding time. Our proposed grouped context structure efficiently enhances decoding speed, albeit with a performance reduction (as observed in V3 and V4). By further incorporating the global hyperprior, the method V5 achieves comparable compression performance, with 160 times faster decoding speed compared to the serial context based entropy model (V3) under the same transform network. The introduction of the residual blocks and self attention blocks in the transform network (V6) significantly improves performance, surpassing the gains achieved with V5, which relies solely on the stacked convolutional layer.

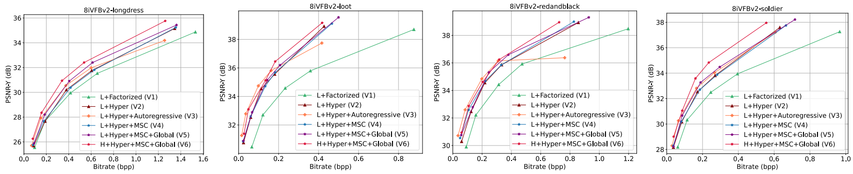

Figure 9: Ablation studies of different entropy models and transform networks on 8iVFBv2 dataset.

Table 3: Complexity and compression performance comparison result of ablation studies (Anchor: L+Factorized).

Methods	BD-PSNR (dB) ↑	BD-Rate (%) ↓	Dec. Time (s) ↓
L+Factorized (V1)	-	-	0.153
L+Hyper (V2)	+1.22	-31.68	0.192
L+Hyper+Autoregressive (V3)	+1.52	-43.80	64.079
L+Hyper+MSC (V4)	+1.24	-32.99	0.222
L+Hyper+MSC+Global (V5)	+1.52	-37.32	0.362
H+Hyper+MSC+Global (V6)	+2.03	-47.07	0.391

6 Conclusion

In this paper, we proposed a lossy parallel point cloud attribute compression scheme, aimed at enhancing decoding speed while maintaining compression performance. Our approach introduces a parallel decoding strategy and a multi-stage context-based entropy model. The parallel decoding strategy involves partitioning the quantized latent feature voxels into non-overlapping groups, allowing for independent context entropy modeling. By integrating short-range and long-range dependencies, ParaPCAC achieves significant decoding speed and quality enhancements in the multi-scale context based entropy model. Experimental results demonstrate that ParaPCAC outperforms existing learning-based methods in terms of rate-distortion performance and decoding latency.

Biographies

Kai Wang received the M.S. degree from the College of Computer Science and Software Engineering, Shenzhen University, China, in 2024. His research interests include point cloud compression.

Pingping Zhang (Student Member, IEEE) received the M.S. degree from the College of Computer Science and Software Engineering, Shenzhen University, China, in 2020. She is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong. Her research interests include learned based image/video compression.

Shengjie Jiao received the B.S. degree in Computer Science and Technology from Wuyi University, China, in 2023. He is currently pursuing the M.S. degree at Shenzhen University. His research interests include point cloud compression.

Hui Yuan (Senior Member, IEEE) received the B.E. and Ph.D. degrees in telecommunication engineering from Xidian University, Xian, China, in 2006 and 2011, respectively. In April 2011, he joined Shandong University, Jinan, China, as a Lecturer (April 2011December 2014), an Associate Professor (January 2015-October 2016), and a Professor (September 2016). From January 2013 to December 2014, and from November 2017 to February 2018, he worked as a Postdoctoral Fellow (Granted by the Hong Kong Scholar Project) and a Research Fellow, respectively, with the Department of Computer Science, City University of Hong Kong. From November 2020 to November 2021, he worked as a Marie Curie Fellow (Granted by the Marie Skodowska-Curie Actions Individual Fellowship under Horizon2020 Europe) with the School of Engineering and Sustainable Development, De Montfort University, Leicester, U.K. From October 2021 to November 2021, he also worked as a visiting researcher (secondment of the Marie Skodowska-Curie Individual Fellowships) with the Computer Vision and Graphics group, Fraunhofer Heinrich-Hertz-Institut (HHI), Germany. His current research interests include 3D visual coding and communication. He served as an Area Chair for IEEE ICME 2023, ICME 2022, ICME 2021, IEEE ICME 2020, and IEEE VCIP 2020. He serves as a member of IEEE CTSoc Audio/Video Systems and Signal Processing Technical Committee (AVS TC) and APSIPA Image, Video, and Multimedia Technical Committee.

Shiqi Wang (Senior Member, IEEE) received the Ph.D. degree in computer application technology from Peking University in 2014. He is currently an Associate Professor with the Department of Computer Science, City University of Hong Kong. He has proposed more than 70 technical proposals to ISO/MPEG, ITU-T, and AVS Standards. He has authored or coauthored more than 300 refereed journal articles/conference papers, including more than 100 IEEE TRANSACTIONS. His research interests include video compression, image/video quality assessment, video coding for machine, and semantic communication. He received the Best Paper Award from IEEE VCIP 2019, ICME 2019, IEEE MULTIMEDIA 2018, and PCM 2017. His coauthored article received the Best Student Paper Award in the IEEE ICIP 2018. He served or serves as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CYBERNETICS, IEEE ACCESS, and APSIPA Transactions on Signal and Information Processing.

Xu Wang (Member, IEEE) received the B.S. degree from South China Normal University, Guangzhou, China, in 2007, the M.S. degree from Ningbo University, Ningbo, China, in 2010, and the Ph.D. degree from the Department of Computer Science, City University of Hong Kong in 2014. In 2015, he joined the College of Computer Science and Software Engineering, Shenzhen University, where he is currently an Associate Professor. His research interests include video coding and 3D vision.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62371310, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011236.

References

[1] E. Alexiou, K. Tung, and T. Ebrahimi, "Towards neural network approaches for point cloud compression", in *Applications of digital image processing XLIII*, Vol. 11510, SPIE, 2020, 18–37.

[2] J. Baert, A. Lagae, and P. Dutré, "Out-of-core construction of sparse voxel octrees", in *Proceedings of the 5th high-performance graphics conference*, 2013, 27–32.

[3] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior", in *International Conference on Learning Representations*, 2018.

[4] S. Biswas, J. Liu, K. Wong, S. Wang, and R. Urtasun, "Muscle: Multi sweep compression of lidar using deep entropy models", *Advances in Neural Information Processing Systems*, 33, 2020, 22170–81.

[5] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves", *ITU SG16 Doc. VCEG-M33*, 2001.

[6] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, 3075–84.

[7] R. A. Cohen, D. Tian, and A. Vetro, "Attribute compression for sparse point clouds using graph transforms", in *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2016, 1374–8.

[8] E. d'Eon, B. Harrison, T. Myers, and P. A. Chou, "8i Voxelized Full Bodies - A Voxelized Point Cloud Dataset", *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) m38673/M72012*, May 2016.

[9] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. NieSSner, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[10] R. L. De Queiroz and P. A. Chou, “Compression of 3D point clouds using a region-adaptive hierarchical transform”, *IEEE Transactions on Image Processing*, 25(8), 2016, 3947–56.

[11] R. L. De Queiroz and P. A. Chou, “Transform coding for point clouds using a Gaussian process model”, *IEEE Transactions on Image Processing*, 26(7), 2017, 3507–17.

[12] G. Fang, Q. Hu, H. Wang, Y. Xu, and Y. Guo, “3dac: Learning attribute compression for point clouds”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 14819–28.

[13] 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*, Vol. 36, 2022, 625–33.

[14] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, “Checkerboard context model for efficient learned image compression”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 14771–80.

[15] Q. Hu, B. Yang, S. Khalid, W. Xiao, N. Trigoni, and A. Markham, “Towards Semantic Segmentation of Urban-Scale 3D Point Clouds: A Dataset, Benchmarks and Challenges”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[16] J.-H. Kim, B. Heo, and J.-S. Lee, “Joint global and local hierarchical priors for learned image compression”, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 5992–6001.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context”, in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, 740–55.

[18] D. Minnen, J. Ballé, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression”, *Advances in neural information processing systems*, 31, 2018.

[19] D. T. Nguyen and A. Kaup, “Lossless Point Cloud Geometry and Attribute Compression Using a Learned Conditional Probability Model”, *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

[20] E. Pavez, B. Girault, A. Ortega, and P. A. Chou, “Region adaptive graph Fourier transform for 3D point clouds”, in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, 2726–30.

[21] K. Perlin, “An image synthesizer”, *ACM Siggraph Computer Graphics*, 19(3), 1985, 287–96.

[22] R. B. Pinheiro, J.-E. Marvie, G. Valenzise, and F. Dufaux, “NF-PCAC: Normalizing Flow based Point Cloud Attribute Compression”, in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2023, 1–5.

[23] Y. Qian, Z. Tan, X. Sun, M. Lin, D. Li, Z. Sun, L. Hao, and R. Jin, “Learning Accurate Entropy Model with Global Reference for Image Compression”, in *International Conference on Learning Representations*, 2021.

[24] M. Quach, G. Valenzise, and F. Dufaux, “Folding-based compression of point cloud attributes”, in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, 3309–13.

[25] M. Quach, G. Valenzise, and F. Dufaux, “Learning convolutional transforms for lossy point cloud geometry compression”, in *2019 IEEE international conference on image processing (ICIP)*, IEEE, 2019, 4320–4.

[26] 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 Computer Vision and Pattern Recognition*, 2021, 6042–51.

[27] M. Rudolph, A. Riemenschneider, and A. Rizk, “Progressive Coding for Deep Learning based Point Cloud Attribute Compression”, in, *MMVE ’24*, Bari, Italy: Association for Computing Machinery, 2024, 78–84, ISBN: 9798400706189, DOI: 10.1145/3652212.3652217, https://doi.org/10.1145/3652212.3652217.

[28] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokua, S. Lasserre, Z. Li, *et al.*, “Emerging MPEG standards for point cloud compression”, *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1), 2018, 133–48.

[29] Y. Shao, Z. Zhang, Z. Li, K. Fan, and G. Li, “Attribute compression of 3D point clouds using Laplacian sparsity optimized graph transform”, in *2017 IEEE Visual Communications and Image Processing (VCIP)*, IEEE, 2017, 1–4.

[30] X. Sheng, L. Li, D. Liu, Z. Xiong, Z. Li, and F. Wu, “Deep-pcac: An end-to-end deep lossy compression framework for point cloud attributes”, *IEEE Transactions on Multimedia*, 24, 2021, 2617–32.

[31] 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, 14368–77.

[32] 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*, 2022.

[33] J. Wang, D. Ding, and Z. Ma, “Lossless Point Cloud Attribute Compression Using Cross-scale, Cross-group, and Cross-color Prediction”, in *2023 Data Compression Conference (DCC)*, IEEE, 2023, 228–37.

[34] J. Wang and Z. Ma, “Sparse tensor-based point cloud attribute compression”, in *2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, IEEE, 2022, 59–64.

[35] J. Wang, H. Zhu, H. Liu, and Z. Ma, “Lossy point cloud geometry compression via end-to-end learning”, *IEEE Transactions on Circuits and Systems for Video Technology*, 31(12), 2021, 4909–23.

[36] Y. Xu, W. Hu, S. Wang, X. Zhang, S. Wang, S. Ma, Z. Guo, and W. Gao, “Predictive generalized graph Fourier transform for attribute compression of dynamic point clouds”, *IEEE Transactions on Circuits and Systems for Video Technology*, 31(5), 2020, 1968–82.

[37] X. Yang, Y. Shao, S. Liu, T. H. Li, and G. Li, “PDE-based Progressive Prediction Framework for Attribute Compression of 3D Point Clouds”, in *Proceedings of the 31st ACM International Conference on Multimedia, MM ’23*, Ottawa ON, Canada: Association for Computing Machinery, 2023, 9271–81, ISBN: 9798400701085, DOI: 10.1145/3581783.3612422, https://doi.org/10.1145/3581783.3612422.

[38] X. Yi, L. Yao, and W. Ziyu, “Owlii Dynamic human mesh sequence dataset”, *ISO/IEC JTC1/SC29/WG11 (MPEG/JPEG) m41658*, 2017.

[39] C. Zhang, D. Florencio, and C. Loop, “Point cloud attribute compression with graph transform”, in *2014 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2014, 2066–70.

[40] J. Zhang, T. Chen, D. Ding, and Z. Ma, “YOGA: Yet Another Geometry-based Point Cloud Compressor”, in *Proceedings of the 31th ACM International Conference on Multimedia*, 2023.

[41] J. Zhang, T. Chen, D. Ding, and Z. Ma, “G-PCC++: Enhanced Geometry-based Point Cloud Compression”, in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, 1352–63.