3DAC: Learning Attribute Compression for Point Clouds

Guangchi Fang[1,2], Qingyong Hu[3], Hanyun Wang[4], Yiling Xu[5], Yulan Guo[1,2,6*]

[1]Sun Yat-sen University, [2]The Shenzhen Campus of Sun Yat-sen University, [3]University of Oxford
[4]Information Engineering University, [5]Shanghai Jiaotong University
[6]National University of Defense Technology

Abstract

We study the problem of attribute compression for large-scale unstructured 3D point clouds. Through an in-depth exploration of the relationships between different encoding steps and different attribute channels, we introduce a deep compression network, termed 3DAC, to explicitly compress the attributes of 3D point clouds and reduce storage usage in this paper. Specifically, the point cloud attributes such as color and reflectance are firstly converted to transform coefficients. We then propose a deep entropy model to model the probabilities of these coefficients by considering information hidden in attribute transforms and previous encoded attributes. Finally, the estimated probabilities are used to further compress these transform coefficients to a final attributes bitstream. Extensive experiments conducted on both indoor and outdoor large-scale open point cloud datasets, including ScanNet and SemanticKITTI, demonstrated the superior compression rates and reconstruction quality of the proposed method.

1. Introduction

As a common 3D data representation, point clouds have been widely used in a variety of real applications such as mixed reality [28], self-driving vehicles [16, 21], and high-resolution mapping [20, 37]. Thanks to the remarkable progress achieved in 3D acquisition, point clouds become increasingly accessible. However, the storage and transmission of massive irregularly sampled points pose a new challenge to existing compression techniques. In particular, along with the raw 3D coordinates of points, the compression of their attributes (*e.g.*, color, reflectance) is also non-trivial[1]. In this regard, we will study effective attribute compression for unstructured point clouds in this paper.

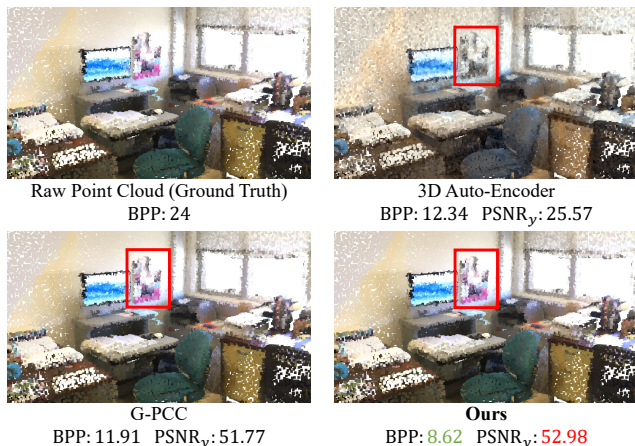To achieve point cloud attribute compression, early

Figure 1. Qualitative point cloud attribute compression results of 3D auto-encoder, G-PCC, and our method on the ScanNet [10]. Bits Per Point (BPP) and Peak Signal-to-Noise Ratio (PSNR) of the luminance component are reported. Note that, the raw point clouds usually use uint8 RGB values (*i.e.*, $8 \times 3 = 24$ BPP).

works [11,29,30,41,42,45,48,57] usually apply image processing techniques to 3D point clouds. In particular, early methods usually follow a two-step framework, *i.e.*, initial coding of attributes and entropy coding of transform coefficients. A number of approaches focus on developing sophisticated initial coding algorithms, which convert point cloud attributes to coefficients in a specific domain. However, entropy coding, which further losslessly encodes coefficients to a final bitstream, has been largely overlooked. There are only a few entropy coders [18,26,39] have been proposed for attribute compression in recent works. In general, entropy coding is independently included in the aforementioned attribute compression framework with initial coding. More specifically, these traditional hand-crafted entropy coders take coefficients as a sequence of symbols and estimate the probability distribution[2] only considering

---

*Corresponding author: Yulan Guo (guoyulan@sysu.edu.cn).

[1]For example, with the standard point cloud compression algorithm of MPEG (*i.e.*, G-PCC [39]), attribute compression takes up to 60% and 90% of the overall bitstream for lossy and lossless compression, respectively.

[2]According to the information theory [40], a lower bound of bitrate can be achieved by an entropy coder given the actual distribution of the transmitted coefficients.

the previous input symbols as context information. Most of prior entropy coders do not incorporate geometry information of point clouds or context information of initial coding. Moreover, these approaches separately encode attributes for each channel and do not make full use of the inter-channel correlations between different attributes.

In this paper, we propose a learning-based compression framework, termed 3DAC, for point cloud attributes. Specifically, the proposed 3DAC adopts Region Adaptive Hierarchical Transform (RAHT) [11] for initial coding. Then, we propose an attribute-oriented deep entropy model to estimate the probability distribution of transform coefficients. In particular, we model the probabilities of these coefficients by exploring context information from the initial coding stage and inter-channel correlations between different attributes. As shown in Fig. 1, our method achieves higher reconstruction quality with a lower bitrate, indicating excellent attribute compression performance. The main contributions of this paper are as follows:

- We introduce a learning-based, effective framework for attribute compression of 3D point clouds, with competitive compression performance.
- We propose an attribute-oriented deep entropy model to connect the initial coding and entropy coding steps in attribute compression, and explore the inter-channel correlations between different attributes.
- We demonstrate the state-of-the-art compression performance of the proposed method on both indoor and outdoor point cloud datasets.

2. Related Work

2.1. Point Cloud Geometry Compression

Point cloud geometry compression aims at compressing the 3D coordinates of points. In light of the unstructured nature of point clouds, existing methods [27, 39] usually leverage an advanced data structure such as octree to organize the raw point clouds. For example, G-PCC [39] includes an octree-based method for geometry compression. Later, in several learning-based approaches [6, 22, 35], the octree structure was also used for initial coding and octree-structured entropy models were proposed for probability estimation. These octree-based approaches are mainly tailored for geometry compression. In addition, deep image compression [3] are extended to the 3D domain [15,23,32,34,50,53] with 3D auto-encoders [7,8,31]. However, due to the high complexity of point cloud attributes, neural networks tend to ignore high-frequency components (*i.e.*, attribute details). Therefore, it is infeasible to extend this framework to attribute compression.

2.2. Point Cloud Attribute Compression

The objective of attribute compression is to compress the attributes (*e.g.*, color, reflectance) of points. The general idea is to apply traditional image compression techniques to 3D point clouds, which usually includes initial coding of attributes and entropy coding of transform coefficients.

Initial coding aims at capturing the signal redundancy in the transform domain. For example, Zhang et al. [57] first split a point cloud into several blocks through octree and used graph transform to convert attributes to eigenvectors. Shao et al. [41,42] organized a point cloud with an KD-Tree and adopted graph transform for attribute compression. Due to the repeated use of eigendecomposition, graph transform-based methods are usually time-consuming. To handle this problem, Queiroz et al. [11] proposed a variation of the Haar wavelet transform, namely, region adaptive hierarchical transform (RAHT), for real-time point cloud transmission. In several follow up works, RAHT is extended with intra-frame and inter-frame prediction [30,45], graph transform [29], and fixed-point operations [38].

Entropy coding tends to encode symbols into a bitstream with the estimated distribution through an entropy coder, such as Huffman [24], arithmetic [52], and Golomb-Rice coders [36, 51]. Zhang et al. [57] and Queiroz et al. [12] assumed that transform coefficients follow a certain Laplacian distribution and then encoded them with arithmetic coding. However, the estimated distribution is usually difficult to approximate the real distribution due to the naive assumption. In [11,30,45], an adaptive run-length Golomb-Rice (RLGR) coder [26] was adopted for entropy coding. Two variations of run-length coding were also adopted by [18, 39]. All these entropy coders take input symbols as sequential data and only adjust their encoding parameters depending on the previous input symbols.

A handful of recent works [6, 33, 43] started to explore deep learning techniques for attribute compression. Maurice et al. [33] used FoldingNet [56] to reorganize a point cloud as an image and then compressed the image with a conventional 2D image codec. Muscle [6] is proposed for dynamic LiDAR intensity compression, which exploited the LiDAR geometry and intensity of the previous frame for its deep entropy model. The compression performance can further be improved by exploiting context information from other initial coding methods and point cloud data structures.

2.3. Deep Image Compression

Traditional image compression methods [5,17,44,46,49] typically contain three steps: transformation, quantization, and entropy coding. In particular, the source image is first transformed from signal domain to frequency domain to capture spatial redundancy with a few transform coefficients. These coefficients are then quantized with a hand-crafted quantization table and entropy coded for further compression.

In contrast to traditional image compression approaches, deep learning based methods jointly optimize the three
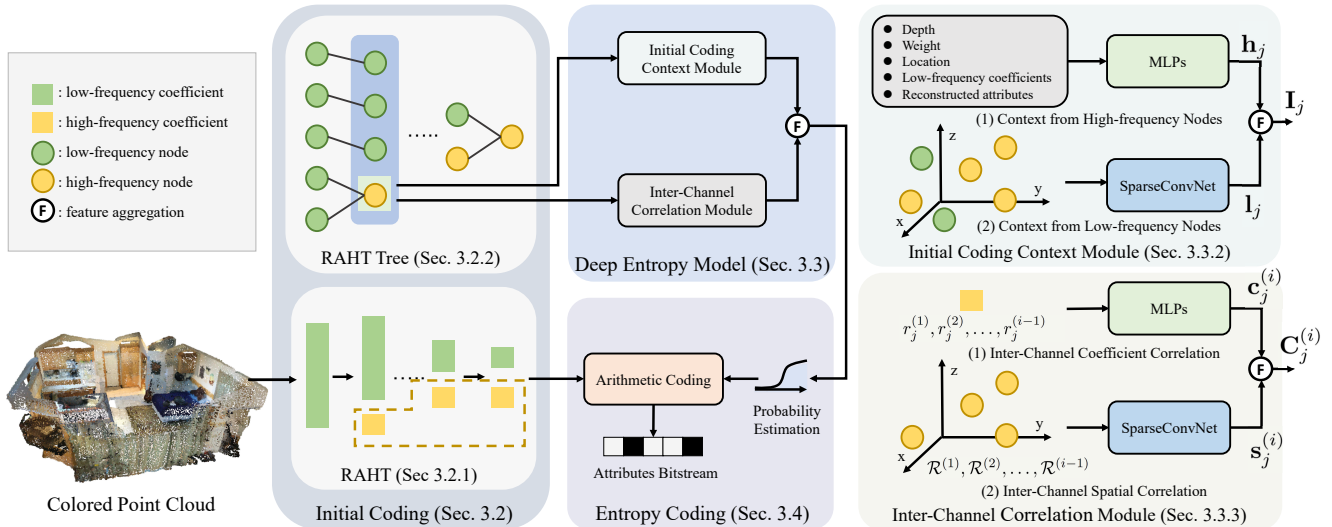
Figure 2. The network architecture of our point cloud attribute compression method.

steps. Most recent methods [1–3, 13, 14, 19, 54] follow an auto-encoder pipeline. These methods formulate a nonlinear function with an encoder to map the source image to a more compressible latent space, and recover the image from the latent codes with a decoder. Then, they also model the entropy of the latent codes with neural networks. For example, Ballé et al. [2] used 2D convolution neural networks to model both the non-linear transform and the entropy model for image compression.

3. Methodology

3.1. Overview

Given a 3D point cloud, its geometry is assumed to have been transmitted separately and this paper mainly focuses on the task of point cloud attribute compression. We propose a learning based attribute compression framework, termed 3DAC, to reduce the storage while ensuring reconstruction quality. Without loss of generality, our framework takes a colored point cloud as its input, as shown in Fig. 2. Note that, other attributes, such as opacity and reflectance, can also be compressed with our framework.

As depicted in Fig. 2, we first adopt an efficient initial coding method (*i.e.*, RAHT) to decompose point cloud attributes into low- and high-frequency coefficients. Then, we model this process with a tree structure (*i.e.*, RAHT tree). The resulting high-frequency coefficients are quantized and formulated as a sequence of symbols via the RAHT tree. Next, we propose an attribute-oriented entropy model to exploit the initial coding context and the inter-channel correlation. Consequently, the probability distribution of each symbol can be well modelled. In the entropy coding stage, the symbol stream and the predicted probabilities are fur-

ther passed into an arithmetic coder to produce the final attributes bitstream.

3.2. Initial Coding

The initial coding methods are capable of capturing spatial redundancy of point cloud attributes by converting attributes to transform coefficients. Here, we adopt Region Adaptive Hierarchical Transform (RAHT) [11] for initial coding due to its simplicity and flexibility. Note that, it is possible to integrate an improved version of RAHT into our framework for better performance.

3.2.1 Region Adaptive Hierarchical Transform

RAHT is a variation of Haar wavelet transform, which converts point cloud attributes to frequency-domain coefficients. Specifically, the raw point cloud is first voxelized, and the attributes are then decomposed to low- and high-frequency coefficients by combining points from individual voxels to the entire 3D space. Here, we briefly revisit RAHT [11] through a 2D toy example.

Figure 3(a) shows an 2D example of RAHT. Only two dimensions (*i.e.*, x and y) are considered in this case. In Fig. 3(a), l_i and h_i denote low- and high-frequency coefficients, respectively. In the first block of Fig. 3(a), points are represented as occupied voxels, and the attributes of corresponding points are denoted as low-frequency coefficients l_1, l_2 and l_3. In the encoding stage, we apply transform along the x axis first and then the y axis by turn until all voxels are merged to the root space. At the first depth level, l_2 and l_3 are transformed to l_4 and h_1, while l_1 is transmitted to the next depth level directly due to lack of neighbor along the x axis. In the second depth level, l_1 and l_4 are transformed
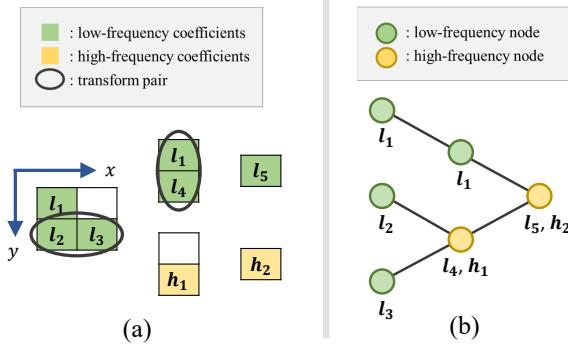
Figure 3. An illustration of RAHT and RAHT tree. (a) A 2D example of RAHT, (b) the corresponding RAHT tree.

to l_5 and h_2 along the y axis. During decoding, the DC coefficient l_5 and high-frequency coefficient h_2 are used to restore low-frequency coefficients l_1 and l_4. Similarly, h_1 is used for l_2 and l_3 with the reconstructed l_4. Thus, only the DC coefficient l_5 and all high-frequency coefficients h_1 and h_2 are required to be transmitted as symbols. That is, these coefficients have to be encoded for attribute compression.

For 3D point clouds, RAHT transforms attributes to co-efficients along three dimensions repeatedly (*e.g.*, along the x axis first, then the y axis and the z axis) until all subspaces are merged to the entire 3D space. Two neighboring points are merged with the following transform:

$$\begin{bmatrix} l_{d+1,x,y,z} \\ h_{d+1,x,y,z} \end{bmatrix} = \mathbf{T}_{w_1,w_2} \begin{bmatrix} l_{d,2x,y,z} \\ l_{d,2x+1,y,z} \end{bmatrix}, \qquad (1)$$

where $l_{d,2x,y,z}$ and $l_{d,2x+1,y,z}$ are low-frequency coefficients of two neighboring points along the x dimension, and $l_{d+1,x,y,z}$ and $h_{d+1,x,y,z}$ are the decomposed low-frequency and high-frequency coefficients. Here, $\mathbf{T}_{w_1}$ is defined as

$$\mathbf{T}_{w_1,w_2} = \frac{1}{\sqrt{w_1 + w_2}} \begin{bmatrix} \sqrt{w_1} & \sqrt{w_2} \\ -\sqrt{w_2} & \sqrt{w_1} \end{bmatrix}, \qquad (2)$$

where w_1 and w_2 are the weights (i.e., the number of leaf nodes) of $l_{d,2x,y,z}$ and $l_{d,2x+1,y,z}$, respectively. Low-frequency coefficients are directly passed to the next level if the point does not have a neighbor.

3.2.2 RAHT Tree

We model RAHT with a tree structure for context feature extraction in Sec. 3.3. In general, points are organized as a RAHT tree according to the hierarchical transform steps. Here, we show a constructed RAHT tree of the aforementioned 2D toy example in Fig. 3(b). Consistent to Sec. 3.2.1, tree nodes of l_2 and l_3 are merged to that of l_4 and h_1, while l_1 and l_4 are merged to l_5 and h_2. In this RAHT tree, leaf nodes represent original voxelized points and internal

nodes represent the corresponding subspace. In particular, the low-frequency coefficient of a RAHT tree node is passed to its parent node if this node has no neighbor along the transform direction. Otherwise, two nodes are merged to their parent. Meanwhile, low- and high-frequency coefficients are generated.

For a better illustration of probability and context modeling in Sec. 3.3, we denote nodes only containing low-frequency coefficients as low-frequency nodes (*i.e.*, green nodes in Fig. 3(b)), and others containing both low- and high-frequency coefficients as high-frequency nodes (*i.e.*, yellow nodes in Fig. 3(b)).

3.2.3 Serialization

We quantize all high-frequency coefficients and serialize these quantized coefficients into a symbol stream with a breadth-first traversal of the RAHT tree. Take the aforementioned 2D toy case as an example, we serialize coefficients of the root level as $\{h_2\}$ and the second level as $\{h_1\}$. The serialization of coefficient is lossless and the attribute distortion only comes from the quantization step. Note that, all context information adopted in Secs. 3.3.2 and 3.3.3 is accessible to our deep entropy model during entropy decoding given the point cloud geometry and the breadth-first traversal format. For the DC coefficient, we transmit it directly.

3.3. Our Deep Entropy Model

According to the information theory [40], given the actual distribution of the transmitted high-frequency coefficients $\mathcal{R}$, a lower bound of bitrate can be achieved by entropy coding. However, the actual distribution $p(\mathcal{R})$ is usually unavailable in practice. To deal with this problem, we propose a deep entropy model to approximate the unknown distribution $p(\mathcal{R})$ with the estimated distribution $q(\mathcal{R})$. In general, we first formulate $q(\mathcal{R})$ with initial coding context and inter-channel correlation, and then utilize a deep neural network (which consists of two modules) to approximate $q(\mathcal{R})$ with $p(\mathcal{R})$ using the cross-entropy loss.

3.3.1 Formulation

Given a 3D point cloud and its associated RAHT tree with m high-frequency nodes and n attribute channels, the transmitted high-frequency coefficients are denoted as $\mathcal{R} = \left\{ \mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(n)} \right\}$ and $\mathcal{R}^{(i)} = \left\{ r_1^{(i)}, \ldots, r_m^{(i)} \right\}$. Considering the inter-channel correlation, we first factorize $q(\mathcal{R})$ into a product of conditional probabilities:

$$q(\mathcal{R}) = \prod_i q\left(\mathcal{R}^{(i)} \mid \mathcal{R}^{(i-1)}, \ldots, \mathcal{R}^{(1)} \right). \qquad (3)$$

Here, the probability distribution of $\mathcal{R}^{(i)}$ is assumed to depend on previous encoded sequences $\left\{ \mathcal{R}^{(1)}, \ldots, \mathcal{R}^{(i-1)} \right\}$.

It is noticed that the probability distribution $q(\mathcal{R})$ also depends on the context information from the initial coding stage, hence we further factorize $q(\mathcal{R}^{(i)})$ with initial coding context $\mathbf{I}_j$ of the coefficient $r_j^{(i)}$:

$$q(\mathcal{R}^{(i)}) = \prod_j q\left(r_j^{(i)} \mid \mathbf{I}_j, \mathcal{R}^{(i-1)}, \dots, \mathcal{R}^{(1)}\right). \quad (4)$$

Then, we model the estimated probability distribution $q\left(\cdot \mid \mathbf{I}_j, \mathcal{R}^{(i-1)}, \dots, \mathcal{R}^{(1)}\right)$ through a probability density model [3] with two proposed context modules, including the initial coding context module (which encodes the information from attribute transform) and the inter-channel correlation module (which explores the dependence on previously encoded attributes).

3.3.2 Initial Coding Context Module

As mentioned in Section 3.2, we adopt RAHT for initial coding and represent the process of RAHT with a tree structure. Here, we exploit the context information hidden in the initial coding stage by extracting context features from both low- and high-frequency tree nodes, and propose our initial coding context module.

Context from High-frequency Nodes. We first start to process the information of high-frequency nodes. As mentioned in Secs. 3.2.2 and 3.2.3, the transmitted symbol stream is composed of high-frequency coefficients, and each coefficient has a corresponding high-frequency node. In light of the strong relationships between the high-frequency coefficients and nodes, our initial coding context module follows [22] to extract latent embedding $\mathbf{h}_j$ of each high-frequency node. In particular, we feed context information, which is obtained in the initial coding stage, to a MultiLayer Perceptron (MLP) to obtain $\mathbf{h}_j$. For a given high-frequency node, the context information contains the depth level, the weight (*i.e.*, the number of child nodes), the low-frequency coefficient, and the attributes. Note that, all information is available during decoding.

Context from Low-frequency Nodes. We further extract the information of low-frequency nodes, and fuse it with that of high-frequency ones into the initial coding context feature. Although low-frequency nodes do not interact directly with the transmitted symbols, there remains a massive quantity of information hidden in these nodes. Thus, we extract the context information that exists in these nodes. Here, we use SparseConv [9, 47] to process RAHT tree nodes for efficiency. In particular, we perform 3D sparse convolutions at each depth level, and take both locations of low- and high-frequency nodes (*i.e.*, center of corresponding subspace) as input points and their context information as input features. With the progressive sparse convolution and downsampling, the context information of low-frequency nodes is fused with that of high-frequency nodes

and diffused into multi-scale feature volumes. For each high-frequency node, we interpolate the latent features from multi-scale feature spaces at its 3D location, and then concatenate them into a final embedding feature $\mathbf{l}_j$.

We concatenate the latent embeddings $\mathbf{h}_j$ and $\mathbf{l}_j$, and feed them into a MLP to obtain the initial coding context $\mathbf{I}_j$ for each transmitted coefficients:

$$\mathbf{I}_j = \mathrm{MLP}([\mathbf{h}_j, \mathbf{l}_j]). \quad (5)$$

3.3.3 Inter-Channel Correlation Module

In most cases, a point cloud contains multiple attribute channels (*e.g.*, color, normal) and there is significant information redundancy between different attributes. Thus, we propose an inter-channel correlation module to exploit the inter-channel correlation.

Inter-Channel Coefficient Correlation. For an uncompressed coefficient $r_j^{(i)}$, we first incorporate the previous encoded coefficients $\left\{r_j^{(1)}, \dots, r_j^{(i-1)}\right\}$ to explore the inter-channel coefficient correlation. Specifically, we use the previous encoded coefficients as prior knowledge and feed them into a MLP to extract the latent feature $\mathbf{c}_j^{(i)}$:

$$\mathbf{c}_j^{(i)} = \mathrm{MLP}([r_j^{(1)}, \dots, r_j^{(i-1)}]). \quad (6)$$

Here, we simply concatenate all previously encoded coefficients and use them as the input of an MLP layer.

Inter-Channel Spatial Correlation. We further aggregate the encoded coefficients through point cloud geometry to benefit the probability estimation. A key observation is that, the attributes of other points are helpful in predicting those of a given point. Therefore, we utilize the spatial relationships via the RAHT tree by incorporating all high-frequency nodes through 3D sparse convolution. Note that, only high-frequency nodes are considered in this part because low-frequency nodes do not contain transmitted coefficients. More specifically, we perform 3D sparse convolutions at each depth level, and take locations of high-frequency nodes as input points and their encoded coefficients as input features. For each high-frequency node, we obtain the coefficient spatial embedding $\mathbf{s}_j^{(i)}$ by interpolating the latent feature in multi-scale feature space.

Similar to the aggregation of initial coding context features, we first concatenate $\mathbf{c}_j^{(i)}$ and $\mathbf{s}_j^{(i)}$, and then feed them into a MLP layer to obtain the prior channel embedding $\mathbf{C}_j^{(i)}$. For the first attribute channel, we set the embedding $\mathbf{C}_j^{(1)}$ as all zeros features for model consistency. Thus, we can finally model $q\left(r_j^{(i)} \mid \mathbf{I}_j, \mathcal{R}^{(i-1)}, \dots, \mathcal{R}^{(1)}\right)$ as $q(r_j^{(i)} \mid \mathbf{I}_j, \mathbf{C}_j^{(i)})$.
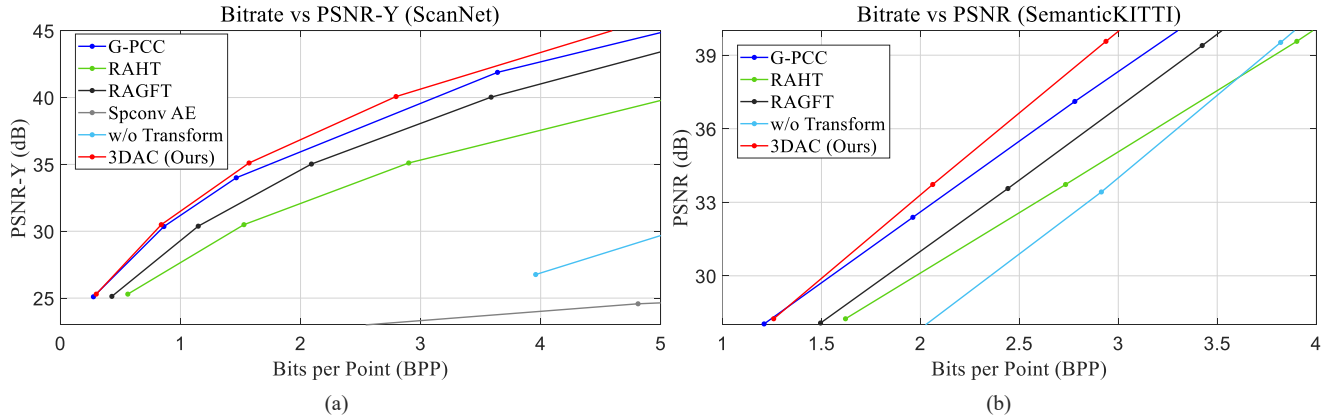
Figure 4. Quantitative results of different attribute compression approaches on the ScanNet (a) and SemanticKITTI (b) datasets.

3.3.4 Probability Estimation

We aggregate two context features $\mathbf{I}_j$ and $\mathbf{C}_j^{(i)}$ through a MLP layer to obtain the final latent embedding. This embedding is further used to generate learnable parameters for a fully factorized density model [3], which is able to model the probability distribution $q\left(r_j^{(i)} \mid \mathbf{I}_j, \mathcal{R}^{(i-1)}, \ldots, \mathcal{R}^{(1)}\right)$.

3.4. Entropy Coding

For entropy coding, we adopt an arithmetic coder to obtain the final attributes bitstream. In the previous steps, we have already converted attributes to transform coefficients in Sec. 3.2.1, serialized these coefficients to a symbol stream in Sec. 3.2.3, and obtained the probabilities of symbols in Sec. 3.3.4. At the final entropy encoding stage, the transform coefficients and the estimated probabilities are passed to the arithmetic coder to generate a final attribute bitstream for further compression. During decoding, the arithmetic coder is capable of restoring the coefficients with the same probability produced by our deep entropy model.

3.5. Learning

During training, we adopt the cross-entropy loss for the deep entropy model:

$$\ell = -\sum_i \sum_j \log q(r_j^{(i)} \mid \mathbf{I}_j, \mathbf{C}_j^{(i)}). \qquad (7)$$

To reduce the bitrate of the final attributes bitstream, we approximate the estimated distribution $q(\mathcal{R})$ with the actual distribution $p(\mathcal{R})$ by minimizing the cross-entropy loss.

4. Experiments

In this section, we first evaluate the attribute compression performance of our method on two point cloud datasets. Then, the effectiveness of our approach is validated on the downstream tasks. We finally conduct extensive ablative experiments to validate the contribution of each component.

4.1. Experimental Setup

(1) Datasets. We conduct experiments on the following two datasets:

- **ScanNet [10].** This is a large-scale indoor point cloud dataset containing 1513 dense point clouds. Each scan contains over ten thousand colored points. Following the official training/testing split, we use 1,201 point clouds for training and 312 point clouds for test.

- **SemanticKITTI [4].** This is a large-scale outdoor Li-DAR dataset with 22 sparse point cloud sequences. Point cloud reflectance captured by LiDAR sensors is used for attribute compression. Following the default setting in [4], we use 11 point sequences for training and the other 11 sequences for test.

(2) Evaluation Metrics. The peak signal-to-noise ratio (PSNR) is reported to evaluate the reconstruction quality. Following [11], we report the peak signal-to-noise ratio of the luminance component in ScanNet. Analogously, the peak signal-to-noise ratio of reflectance is reported on the SemanticKITTI dataset. Bits Per Point (BPP) is also adopted to evaluate the compression ratio.

(3) Baselines. We compare the proposed method with the following selected baselines:

- **RAHT [11].** We use RAHT for initial coding and the run-length Golomb-Rice coder [26] for entropy coding.

- **RAGFT [29].** This is an improved version of RAHT with graph transform. The run-length Golomb-Rice coder is also adopted for entropy coding.

- **G-PCC [39].** This is a standard point cloud compression method (**G-PCC**) provided by MPEG[3].

- **Spconv AE.** We adopt torchsparse [47] to construct MinkUNet [9] for attribute reconstruction and use a fully factorized density model [3] for entropy coding.

---

[3]We use version 13.0 of G-PCC reference implementation: https://github.com/MPEGGroup/mpeg-pcc-tmc13
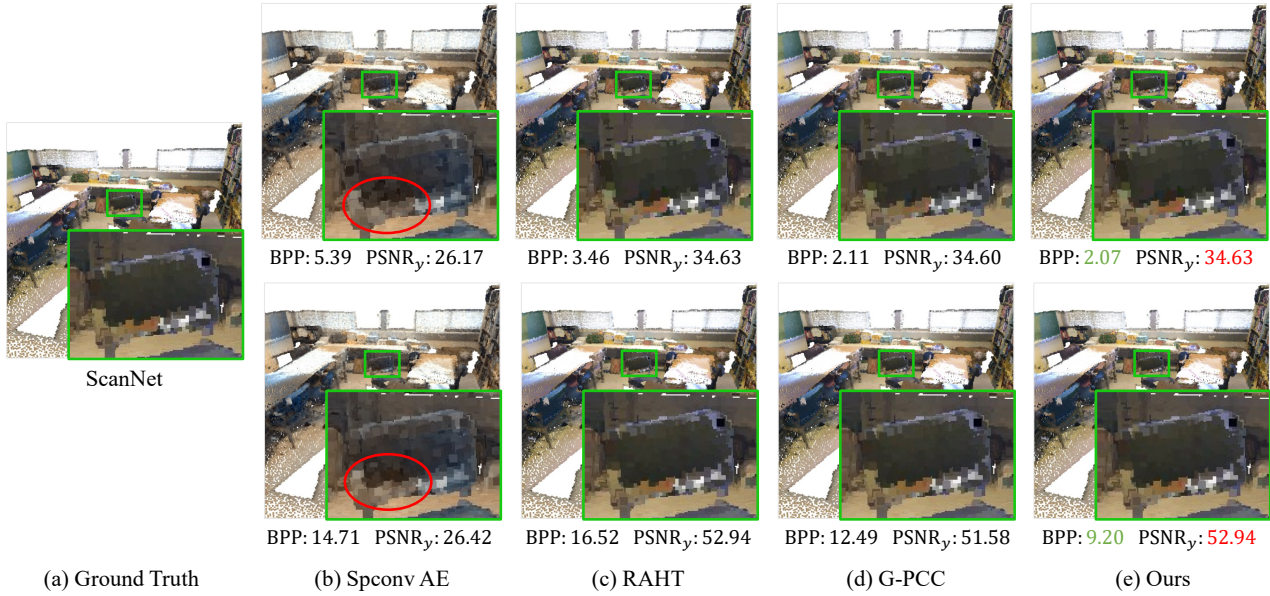
Figure 5. Qualitative results achieved by our method and other baselines including Spconv AE, RAHT [11], and G-PCC [39]. We visualize ScanNet scans at low and high bitrates, respectively. Our method achieves the best compression quality (PSNR) with the lowest bitrates.

- **w/o Transform**. This is motivated by MuSCLE [6]. The attributes are transmitted without any transform and entropy coded by a probability density model [3] with geometric information. In contrast to [6], we extract geometric context features through 3D sparse convolution from the current point cloud frame.

Note that, uniform quantization of transform coefficients is used in our method, RAHT and RAGFT, while adaptive quantization is used in G-PCC.

Implementation Details. In order to simulate the real condition of point cloud compression, we voxelize the raw point cloud data with the 9-level and 12-level octree for ScanNet and SemanticKITTI, respectively, and assume that the geometry of point clouds has been transmitted separately. For ScanNet, a conversion is performed from the RGB color space to the YUV color space following the default setting of G-PCC [39]. We adopt both of the initial coding context module and the inter-channel correlation module. For SemanticKITTI, considering that a single channel of attribute (*i.e.*, reflectance) is included, we only use the initial coding context module and simply disable the inter-channel correlation module.

4.2. Evaluation on Public Datasets

Evaluation on ScanNet. The quantitative attribute compression results of different approaches on ScanNet are shown in Fig. 4. It can be seen that transform-based methods (*i.e.*, 3DAC, G-PCC, RAHT, and RAGFT) significantly outperform other methods (*i.e.*, w/o Transform and Spconv

AE), demonstrating the effectiveness of the initial coding scheme. Additionally, the proposed method consistently achieves the best results compared with other baselines. Although the same initial coding is adopted for our method and RAHT, our method can achieve the same reconstruction quality with a much lower bitrate. This can be mainly attributed to the proposed entropy model and the learning framework. We also provide the qualitative comparison on ScanNet in Fig. 5. It can be seen that our approach achieves better reconstruction performance even compared with the standard point cloud compression algorithm, G-PCC.

Evaluation on SemanticKITTI. The quantitative result of the compression performance achieved by different methods on the SemanticKITTI dataset is shown in Fig. 4. It is clear that the proposed method consistently outperforms other methods by a large margin. This is primarily because that the proposed module is able to implicitly learn the geometry information from sparse LiDAR point cloud through 3D convolutions. We also noticed that existing compression baselines (including RAHT, RAGFT, and G-PCC) achieve unsatisfactory performance on this dataset, since these methods are mainly developed for dense point clouds. In contrast, the proposed method is demonstrated to work well on both indoor dense point clouds and outdoor sparse LiDAR point clouds.

4.3. Evaluation on Downstream Tasks

We further evaluate the performance of attribute compression through two representative downstream tasks, including a 3D scene flow estimation task for machine per-
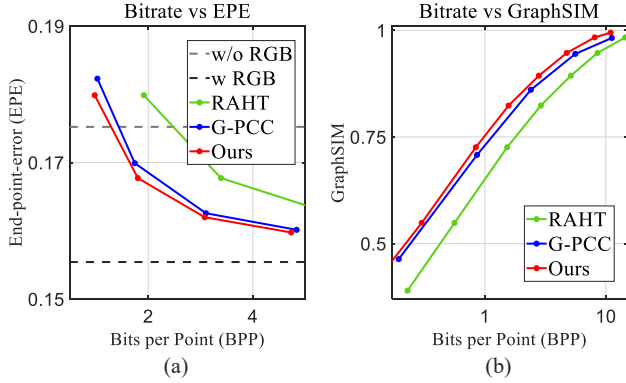
Figure 6. The quantitative results on two downstream tasks. (a): Scene flow estimation on FlyingThings3D. (b): Quality asessment on ScanNet.

ception and a quality assessment task for human perception.

Scene Flow Estimation. We present the performance of scene flow estimation on FlyingThings3D with FlowNet3D [25]. In particular, we first train the network on the raw point clouds. Then, we evaluate the flow estimation performance on the compressed point clouds. The results of raw point clouds with and without attributes are also reported as 'w RGB' and 'w/o RGB', respectively. End-point-error (EPE) at different bits per point is adopted as the evaluation metric. As shown in Fig. 6, the proposed method achieves better performance than RAHT and G-PCC. The performance on this downstream task further demonstrates the superior compression quality of our method.

Quality Assessment. For human perception tasks, we adopt point cloud quality assessment here since the human visual system is more sensitive to attributes. In particular, we adopt the GraphSIM [55] metric to indicate the quality of the compressed point clouds, and higher GraphSIM score means lower attribute distortion. The scores at different bits per point are reported. As shown in Fig. 6, the proposed method achieves the highest GraphSIM score at the same bitrate, further demonstrating the effectiveness of our approach on downstream human vision-oriented tasks.

Initial Coding		Inter Channel		Birate
H	L	C	S	
				3.85
✓				3.48
✓	✓			3.28
✓	✓	✓		3.08
✓	✓	✓	✓	2.79

Table 1. Ablation study on the initial coding context and inter-channel correlation modules. H, L, C, and S denote the information from High-frequency nodes, information from Low-frequency nodes, inter-channel Coefficient dependence, and inter-channel Spatial dependence, respectively.

D	W	F	A	Birate
				3.85
✓				3.64
✓	✓			3.57
✓	✓	✓		3.51
✓	✓	✓	✓	3.48

Table 2. Ablation study on information from high-frequency nodes. D, W, F, and A stand for the node's RAHT tree depth level, weight, low-frequency coefficients, and attributes, respectively.

4.4. Ablation Study

To further determine the contribution of each component in our framework, we conduct several groups of ablation experiments in this section. Note that, all experiments are conducted on ScanNet and all uniform quantization parameters are set as 10 for the same reconstruction quality. First, we evaluate the initial coding context and inter-channel correlation modules. As shown in Table 1, we start with using a non-parametric, fully factorized density model [3], and then add information from high- and low-frequency nodes (**H** and **L**, respectively), inter-channel coefficient dependence (**C**), and finally adopt inter-channel spatial dependence (**S**). It is clear that the progressive incorporation of the proposed components can lead to a lower bitrate.

We then report the ablation studies over the context information of the RAHT-tree node in Table 2. By progressively incorporating the node's depth level (**D**), weight (**W**), low-frequency coefficients (**F**), and attributes (**A**) into the context information of the high-frequency nodes, we can see a steady reduction of the encoding entropy, verifying the effectiveness of the proposed components.

5. Conclusion

In this paper, we presented a point cloud attribute compression algorithm. Our method includes an attribute-oriented deep entropy model considering both attribute initial coding and inter-channel correlations to reduce the storage of attributes. We showed the compression performance of our method over both indoor and outdoor datasets, and the results demonstrated that our approach has superior capability to reduce the bitrate as well as ensuring the reconstruction quality.

References

[1] Yuanchao Bai, Xianming Liu, Wangmeng Zuo, Yaowei Wang, and Xiangyang Ji. Learning scalable ly=-constrained near-lossless image compression via joint lossy image and residual compression. In CVPR, pages 11946–11955, 2021. 3

[2] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. In ICLR, 2017. 3

[3] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In ICLR, 2018. 2, 3, 5, 6, 7, 8

[4] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In ICCV, pages 9297–9307, 2019. 6

[5] Fabrice Bellard. Bpg image format. https://bellard.org/bpg, 2015. 2

[6] Sourav Biswas, Jerry Liu, Kelvin Wong, Shenlong Wang, and Raquel Urtasun. Muscle: Multi sweep compression of lidar using deep entropy models. NeurIPS, 33, 2020. 2, 7

[7] Andrew Brock, Theodore Lim, James Millar Ritchie, and Nicholas J Weston. Generative and discriminative voxel modeling with convolutional neural networks. In Neural Inofrmation Processing Conference: 3D Deep Learning, 2016. 2

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In CVPR, pages 3070–3079. IEEE, 2019. 2

[9] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D spatio-temporal convnets: Minkowski convolutional neural networks. In CVPR, pages 3075–3084, 2019. 5, 6

[10] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In CVPR, pages 5828–5839, 2017. 1, 6

[11] Ricardo L De Queiroz and Philip A Chou. Compression of 3d point clouds using a region-adaptive hierarchical transform. IEEE TIP, 25(8):3947–3956, 2016. 1, 2, 3, 6, 7

[12] Ricardo L de Queiroz and Philip A Chou. Transform coding for point clouds using a gaussian process model. IEEE TIP, 26(7):3507–3517, 2017. 2

[13] Xin Deng, Wenzhe Yang, Ren Yang, Mai Xu, Enpeng Liu, Qianhan Feng, and Radu Timofte. Deep homography for efficient stereo image compression. In CVPR, pages 1492–1501, 2021. 3

[14] Ge Gao, Pei You, Rong Pan, Shunyuan Han, Yuanyuan Zhang, Yuchao Dai, and Hojae Lee. Neural image compression via attentional multi-scale back projection and frequency decomposition. In ICCV, pages 14677–14686, 2021. 3

[15] Linyao Gao, Tingyu Fan, Jianqiang Wan, Yiling Xu, Jun Sun, and Zhan Ma. Point cloud geometry compression via neural graph sampling. In ICIP, pages 3373–3377. IEEE, 2021. 2

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In CVPR, pages 3354–3361, 2012. 1

[17] Vivek K Goyal. Theoretical foundations of transform coding. IEEE Signal Processing Magazine, 18(5):9–21, 2001. 2

[18] Shuai Gu, Junhui Hou, Huanqiang Zeng, Hui Yuan, and Kai-Kuang Ma. 3d point cloud attribute compression using geometry-guided sparse representation. IEEE TIP, 29:796–808, 2019. 1, 2

[19] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. Checkerboard context model for efficient learned image compression. In CVPR, pages 14771–14780, 2021. 3

[20] Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, and Andrew Markham. Sensaturban: Learning semantics from urban-scale photogrammetric point clouds. International Journal of Computer Vision, pages 1–28, 2022. 1

[21] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Learning semantic segmentation of large-scale point clouds with random sampling. IEEE TPAMI, 2021. 1

[22] Lila Huang, Shenlong Wang, Kelvin Wong, Jerry Liu, and Raquel Urtasun. Octsqueeze: Octree-structured entropy model for lidar compression. In CVPR, pages 1313–1323, 2020. 2, 5

[23] Tianxin Huang and Yong Liu. 3d point cloud geometry compression on deep learning. In ACM MM, pages 890–898, 2019. 2

[24] David A Huffman. A method for the construction of minimum-redundancy codes. Proceedings of the IRE, 40(9):1098–1101, 1952. 2

[25] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In CVPR, pages 529–537, 2019. 8

[26] Henrique S Malvar. Adaptive run-length/golomb-rice encoding of quantized generalized gaussian sources with unknown statistics. In Data Compression Conference (DCC'06), pages 23–32. IEEE, 2006. 1, 2, 6

[27] Donald Meagher. Geometric modeling using octree encoding. Computer graphics and image processing, 19(2):129–147, 1982. 2

[28] Zhigeng Pan, Adrian David Cheok, Hongwei Yang, Jiejie Zhu, and Jiaoying Shi. Virtual reality and mixed reality for virtual learning environments. Computers & graphics, 30(1):20–28, 2006. 1

[29] Eduardo Pavez, Benjamin Girault, Antonio Ortega, and Philip A Chou. Region adaptive graph fourier transform for 3d point clouds. In ICIP, pages 2726–2730. IEEE, 2020. 1, 2, 6

[30] Eduardo Pavez, Andre L Souto, Ricardo L De Queiroz, and Antonio Ortega. Multi-resolution intra-predictive coding of 3d point cloud attributes. arXiv preprint arXiv:2106.08562, 2021. 1, 2

[31] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In CVPR, pages 652–660, 2017. 2

[32] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux. Learning convolutional transforms for lossy point cloud geometry compression. In ICIP, pages 4320–4324. IEEE, 2019. 2

[33] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux. Folding-based compression of point cloud attributes. In ICIP, pages 3309–3313. IEEE, 2020. 2

[34] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux. Improved deep point cloud geometry compression. In 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), pages 1–6. IEEE, 2020. 2

[35] Zizheng Que, Guo Lu, and Dong Xu. Voxelcontext-net: An octree based framework for point cloud compression. In CVPR, pages 6042–6051, 2021. 2

[36] Iain E Richardson. H. 264 and MPEG-4 video compression: video coding for next-generation multimedia. John Wiley & Sons, 2004. 2

[37] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. Robotics and Autonomous Systems, 2008. 1

[38] Gustavo P Sandri, Philip A Chou, Maja Krivokuća, and Ricardo L de Queiroz. Integer alternative for the region-adaptive hierarchical transform. IEEE Sign. Process. Letters, 26(9):1369–1372, 2019. 2

[39] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A Chou, Robert A Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, et al. Emerging mpeg standards for point cloud compression. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 9(1):133–148, 2018. 1, 2, 6, 7

[40] Claude Elwood Shannon. A mathematical theory of communication. The Bell system technical journal, 27(3):379–423, 1948. 1, 4

[41] Yiting Shao, Qi Zhang, Ge Li, Zhu Li, and Li Li. Hybrid point cloud attribute compression using slice-based layered structure and block-based intra prediction. In ACM MM, pages 1199–1207, 2018. 1, 2

[42] Yiting Shao, Zhaobin Zhang, Zhu Li, Kui Fan, and Ge Li. Attribute compression of 3d point clouds using laplacian sparsity optimized graph transform. In VCIP, pages 1–4. IEEE, 2017. 1, 2

[43] Xihua Sheng, Li Li, Dong Liu, Zhiwei Xiong, Zhu Li, and Feng Wu. Deep-pcac: An end-to-end deep lossy compression framework for point cloud attributes. IEEE TMM, 2021. 2

[44] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The jpeg 2000 still image compression standard. IEEE Signal processing magazine, 18(5):36–58, 2001. 2

[45] André L Souto and Ricardo L de Queiroz. On predictive raht for dynamic point cloud coding. In ICIP, pages 2701–2705. IEEE, 2020. 1, 2

[46] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. IEEE TCSVT, 22(12):1649–1668, 2012. 2

[47] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In ECCV, pages 685–702. Springer, 2020. 5, 6

[48] Dorina Thanou, Philip A Chou, and Pascal Frossard. Graph-based motion estimation and compensation for dynamic 3d point cloud compression. In ICIP, pages 3235–3239. IEEE, 2015. 1

[49] Gregory K Wallace. The jpeg still picture compression standard. IEEE transactions on consumer electronics, 38(1):xviii–xxxiv, 1992. 2

[50] Jianqiang Wang, Dandan Ding, Zhu Li, and Zhan Ma. Multiscale point cloud geometry compression. In 2021 Data Compression Conference (DCC), pages 73–82. IEEE, 2021. 2

[51] Marcelo J Weinberger, Gadiel Seroussi, and Guillermo Sapiro. The loco-i lossless image compression algorithm: Principles and standardization into jpeg-ls. IEEE TIP, 9(8):1309–1324, 2000. 2

[52] Ian H Witten, Radford M Neal, and John G Cleary. Arithmetic coding for data compression. Communications of the ACM, 30(6):520–540, 1987. 2

[53] Wei Yan, Shan Liu, Thomas H Li, Zhu Li, Ge Li, et al. Deep autoencoder-based lossy geometry compression for point clouds. arXiv preprint arXiv:1905.03691, 2019. 2

[54] Fei Yang, Luis Herranz, Yongmei Cheng, and Mikhail G Mozerov. Slimmable compressive autoencoders for practical neural image compression. In CVPR, pages 4998–5007, 2021. 3

[55] Qi Yang, Zhan Ma, Yiling Xu, Zhu Li, and Jun Sun. Inferring point cloud quality via graph similarity. IEEE TPAMI, 2020. 8

[56] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In CVPR, pages 206–215, 2018. 2

[57] Cha Zhang, Dinei Florencio, and Charles Loop. Point cloud attribute compression with graph transform. In ICIP, pages 2066–2070. IEEE, 2014. 1, 2