applied sciences

MDPI

Article

Variable Rate Point Cloud Attribute Compression with Non-Local Attention Optimization

Xiao Huo, Saiping Zhang and Fuzheng Yang *

The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
* Correspondence: fzhyang@mail.xidian.edu.cn

Abstract: Point clouds are widely used as representations of 3D objects and scenes in a number of applications, including virtual and mixed reality, autonomous driving, antiques reconstruction. To reduce the cost for transmitting and storing such data, this paper proposes an end-to-end learning-based point cloud attribute compression (PCAC) approach. The proposed network adopts a sparse convolution-based variational autoencoder (VAE) structure to compress the color attribute of point clouds. Considering the difficulty of stacked convolution operations in capturing long range dependencies, the attention mechanism is incorporated in which a non-local attention module is developed to capture the local and global correlations in both spatial and channel dimensions. Towards the practical application, an additional modulation network is offered to achieve the variable rate compression purpose in a single network, avoiding the memory cost of storing multiple networks for multiple bitrates. Our proposed method achieves state-of-the-art compression performance compared to other existing learning-based methods and further reduces the gap with the latest MPEG G-PCC reference software TMC13 version 14.

Keywords: point cloud; compression; non-local attention mechanism; variable rate model; sparse convolution

check for updates

Citation: Huo, X.; Zhang, S.; Yang, F. Variable Rate Point Cloud Attribute Compression with Non-Local Attention Optimization. *Appl. Sci.* **2022**, *12*, 8179. https://doi.org/10.3390/app12168179

Academic Editor: Dimitris Mourtzis

Received: 22 July 2022
Accepted: 13 August 2022
Published: 16 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Recent years have witnessed tremendous development of 3D capture devices and reconstruction technologies, and point clouds have become a crucial data structure for storage and transmission of 3D data. A point cloud is an unordered set of points in 3D space, usually represented by their coordinates, corresponding to the geometry, and each point may have its associated attributes, such as color and reflectance. Through these properties it can practically represent arbitrary 3D objects or scenes. As more and more point cloud content is made available and non-compressed point cloud requires a huge volume of data, the development of appropriate coding solutions becomes essential.

Motivated by the huge success of deep neural networks (DNN)-based image and video compression [1–4], several studies have been devoted to the use of deep learning for PCAC. Generally, they follow the VAE framework of image compression but in different point cloud representations to perform the attribute compression, such as the point-based PointNet series [5,6] architecture in [7] (Deep-PCAC), voxel-based 3D dense convolutions in [8] and sparse convolution in [9] (SparsePCAC). Specially, Quach et al. [10] builds a mapping between 3D point clouds and 2D grids, and utilizes the existing image compression method to encode and decode the 2D grids. However, the performance of these learning-based solutions still lags behind that of traditional G-PCC. How to develop the potential of learning-based approaches to outperform the traditional methods is still a challenging task.

Point cloud compression can be divided into two main categories: geometry compression and attribute compression. Geometry compression involves compression of point

coordinates in 3D space while attribute compression involves compression of point attributes. In this paper, we focus on point cloud attribute compression and assume the geometry information is known. Without loss of generality, we assume that the attributes are colors represented as (R, G, B) triplets in RGB color space. Considering the efficiency of sparse convolution processing sparsely distributed point clouds, we establish a PCAC framework on top of the sparse convolution [11] based VAE structure. It should be noted that, in the proposed framework, we overcome the defect of other learning-based methods in capturing the long range dependencies in the use of stacked convolution operations. The non-local attention mechanism module (NLAM) is introduced into the proposed VAE structure to capture both local and global correlations among voxels. The non-local attention masks are applied at different layers to distinguish the importance of latent features in both positions and channels dimensions for better compression. Furthermore, in the existing learning-based PCAC methods, the network parameters are learned by jointly minimizing bitrate and distortion at a particular trade-off. To this end each trade-off needs an independent model. Multiple independent models require large memory to store and transmission, which puts great limitations on the practical applications. Inspired by the variable bitrate compression methods of image compression [12–14], we propose a modulation network to adaptively modify the latent features of different layers. This modulation network is based on parameter sharing multilayer perceptron (MLP) and the modification is implemented by simple yet effective channel-wise production. Experimental results have shown that the proposed lightweight but effective modulation network successfully achieve variable bit compression in a single network with only acceptable performance loss.

We evaluate the performance on voxelized 8i Voxelized Full Bodies (8iVFB) [15] which has been adopted by MPEG as the standard test set and experiments are executed under common test conditions. Our proposed method achieves the state-of-the-art coding performance compared with other existing learning-based methods learning-based methods and further reduces the performance gap with the latest MPEG G-PCC coding software TMC13 version 14 (TMC13v14). Specifically, for the Bjøntegaard Delta rate (BD-rate), the proposed method outperforms Deep-PCAC and SparsePCAC by 50.7% reduction and 6.3% reduction, respectively, and reduces the gap with TMC13v14 to 29.3%.

The primary contributions of this paper include:

(1) This paper explores the application of attention mechanism in point cloud attribute compression. The attention mechanism is integrated in which a non-local attention module is developed to capture the local and global correlations in both spatial and channel dimensions, which is demonstrated to improve the performance of current VAE coding framework.

(2) For practical applications, the proposed modulation network adaptively modifies the latent features of different sparse convolutional layers in a single network, which avoids memory cost of storing multiple networks for multiple bitrates.

(3) We establish an end-to-end learning-based PCAC framework which achieves the state-of-the-art compression performance compared to other existing learning-based methods and further reduces the gap with the latest MPEG-PCC reference software TMC13 version 14.

The remainder of this paper is organized as follows. Section 2 gives a brief review of related works. Section 3 describes the framework of our proposed point cloud attribute autoencoder, the non-local attention module, and the variable rate autoencoder network. Section 4 first depicts the training and testing details. Then, this section presents objective quality comparison results, subjective quality comparison results, and ablation studies. Finally, concluding remarks and future works are described in Section 5.

2. Related Works

In this section, we mainly review point cloud attribute compression including traditional methods and learning-based methods. In addition, we briefly review the successful applications of attention mechanism in image-related tasks.

2.1. Point Cloud Attribute Compression

Zhang et al. [16] first proposed a graph transform-based method to compress attributes, which were constructed of sub-graphs and associated Laplacian matrices to effectively code the colors of a given point cloud. However, it needed to repeatedly solve the eigen-decompositions of many large graph Laplacians, resulting in the approach infeasible for real-time processing. Queiroz and Chou [17] proposed a Region-Adaptive Hierarchical Transform (RAHT) with low complexity. They compressed the colors in a hierarchical sub-band transform that resembled an adaptive variation of a Haar wavelet, and the RAHT coefficients are encoded via arithmetic coding with assumed Laplace distribution. RAHT has been adopted into the MPEG G-PCC [18] as one of the core transformation parts for attribute compression. G-PCC further improved it with better entropy coding in version 6 (TMC13v6) and with the prediction of RAHT coefficients in the latest TMC13v14, which significantly improved performance and achieved the state-of-the-art PCAC efficiency.

In addition to these traditional methods represented by G-PCC, a few pioneering works based on deep learning have emerged recently. Sheng et al. [7] proposed a PointNet [5] and PointNet++ [6] based autoencoder for attribute compression, and simultaneously proposed the improved module second-order point convolutional layer and dense point-inception block, which further extended the local receptive field and improved the ability of feature extraction. However, since PointNet series networks used feature extraction function max() at the cost of information loss, its final compression performance was unsatisfactory. Quach et al. [10] proposed learning a neural network(NN)-based bidirectional mapping between a 3D point cloud and a 2D grid, and then used conventional image codec to encode the generated 2D grid. For every single point cloud, it overfitted the mapping function. However, since wrong mapping and repeated mapping can not be solved well, its final compression performance still suffers and also the model generalization is restricted. Alexiou et al. [8] applied 3D dense convolutional autoencoder for the coding of both geometry and attribute components. In contrast, Wang et al. [9] applied sparse convolution for feature extraction because of its superior efficiency for representing the geometry of unorganized points. They stacked sparse convolutional layers in the VAE structure and used conditional entropy model based on hyperprior and autoregressive prior, which has obviously reduced the gap between the learning-based method and the traditional G-PCC. Following this basic VAE framework, in this paper, we introduce the non-local attention mechanism to further narrow the gap.

2.2. Attention Mechanism

The attention mechanism was proposed in deep learning-based natural language processing (NLP) [19]. Generally, attention mechanism generates importance masks for the input, so as to guide the network to allocate more processing resources towards the most informative part. In recent years, it has also shown the potential to become a dominant tool in computer vision. Non-local attention in deep Convolutional Neural Networks (CNNs) allows the network to concentrate more on noticeable areas. Wang et al. [20] took the lead in introducing self-attention to computer vision and presented a novel non-local network with great success in video understanding and object detection. In image compression, Cheng et al. [21] used discretized Gaussian mixture likelihoods along with attention models to reduce the spatial redundancy after quantization operations. Xue and Su [22] proposed a post-processing-based neural network containing spatial and channel attention modules connected in parallel. However, the application of the attention mechanism in point cloud attribute compression has not been explored. Our proposed model first adopts the attention mechanism in neural network for voxelized point cloud compression. The attention mechanism modules are placed at the shallow layer and bottleneck layer to implicitly and adaptively allocate more bits to more representative latent features for compression.

3. Point Cloud Attribute Variational Autoencoder

Due to the good exploration of sparse convolution on the sparsity of point cloud, we use it as the basic layer of the proposed neural network. It only computes outputs on predefined voxels and saves them into a compact sparse tensor. The sparse tensor is represented by a set of coordinates $\vec{\mathbf{C}} = \{(x_i, y_i, z_i)\}_i$ of these voxels and corresponding features $\vec{\mathbf{F}} = \left\{\vec{f}_i\right\}_i$ which can be written in a simple form:

$$\vec{\mathbf{C}} = \begin{bmatrix} x_1 & y_1 & z_1 \\ & \vdots & \\ x_N & y_N & z_N \end{bmatrix}, \vec{\mathbf{F}} = \begin{bmatrix} f_{11} & \cdots & f_{1C} \\ \vdots & \ddots & \vdots \\ f_{N1} & \cdots & f_{NC} \end{bmatrix} = \begin{bmatrix} \vec{f}_1^T \\ \vdots \\ \vec{f}_N^T \end{bmatrix}, \tag{1}$$

where N is the number of point/feature, and C is the number of channel. Then, sparse convolution can be defined as follows:

$$\vec{f}_u^{\text{out}} = \sum_{i \in K^3\left(u, \vec{\mathbf{C}}^{in}\right)} W_i \vec{f}_{u+i}^{in} \quad for \quad u \in \vec{\mathbf{C}}^{out}, \tag{2}$$

where $\vec{\mathbf{C}}^{in}$ and $\vec{\mathbf{C}}^{out}$ are input and output coordinates, respectively. $\vec{f}_u^{in}$ and $\vec{f}_u^{out}$ are input and output feature vectors at coordinate u (i.e., (x_u, y_u, z_u)), respectively. $K^3\left(u, \vec{\mathbf{C}}^{in}\right) = \left\{i \mid u + i \in \vec{\mathbf{C}}^{in}, i \in K^3\right\}$ defines a 3D convolutional kernel centered at u with offset i in $\vec{\mathbf{C}}^{in}$. W_i is kernel weights.

In 3D dense convolution, the convolution is implemented on each voxel in the 3D dense grid. If the centered voxel is empty and its 3D kernel covers any occupied voxel, the centered voxel will be filled by the value generated by the convolution, which leads to the occupied voxels gradually increase. In contrast, the sparse convolution works strictly on submanifolds of data, hence the generated values also only exist on the submanifolds. Therefore, after deep convolution layers, the point cloud structure could still remain the sparsity, avoiding the difficulty of compression caused by feature map dilation.

3.1. General Framework

Figure 1 illustrates the detailed proposed VAE architecture. It is established on a VAE structure [23], with non-local attention modules in main encoder-decoder pairs. As the geometric information has been assumed decoded, the coordinates $\vec{\mathbf{C}}^{in}$ and $\vec{\mathbf{C}}^{out}$ in Equation (2) have been already known in encoder and decoder. X and $\hat{X}$ denote the feature part of the sparse tensor, which are the input attribute and the reconstructed output color attribute of the point cloud, respectively.

Main encoder with quantization Q is used to generate quantized latent features $\hat{Y}$ and main decoder decodes the features into the reconstructed point cloud. The main encoder consists of Sparse Convolutional (SConv) layers, followed Rectified Linear Unit (ReLU) activation layers and NLAMs. Note that the first convolutional layer has a kernel size of $5 \times 5 \times 5$ to cover more occupied voxels and the left SConv layers have the kernel size of $3 \times 3 \times 3$ for deep features aggregation. The first NLAM is placed at a relatively shallow layer to capture the correlations in lower-level feature map and the second NLAM is placed at the bottleneck layer to intelligently allocate more bits in more informative areas. The structure of the main decoder is the mirror inverse of that of the main encoder. The hyper encoder generates much smaller side information as hyperpriors $\hat{Z}$. $\hat{Z}$ are then processed through the hyper decoder to generate the mean and location parameters (μ, σ) of assumed conditional Laplacian distribution of $\hat{Y}$. Quantization and entropy coding are used to connect the encoder and decoder. The uniform noise approximation [1] and rounding are used in the training and testing phases, respectively. Entropy coding and decoding are used to compress and decompress the quantized hyperpriors and quantized latent features, respectively.

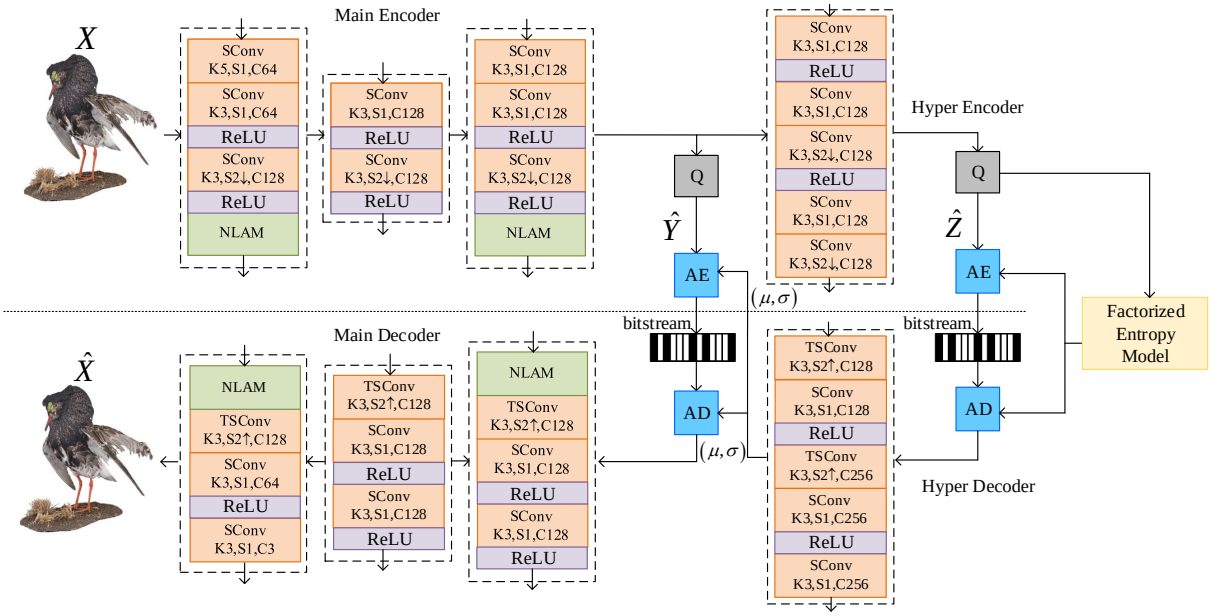


Figure 1. The proposed autoencoder network with non-local attention mechanism module. "SConv K3, S2↓, C128" indicates a downsampling sparse convolution layer with a kernel of size $3 \times 3 \times 3$, stride of 2, and 128 output channels. " TSConv " with "S2↑" stands for upsampling transposed sparse convolution layers with stride of 2. "ReLU" represents the Rectified Linear Unit. NLAM refers to non-local attention modules. "Q" is for quantization. "AE" and "AD" are arithmetic encoding and decoding.

Ideally, we want to use as few bits as possible to represent $\hat{Y}$ and $\hat{Z}$ while minimizing the distortion measurement between X and $\hat{X}$. More specifically, we wish to learn appropriate parameters of main encoder and decoder, and conditional entropy coding for better compression efficiency by minimizing the Lagrangian cost $J = R_{\hat{Y}} + R_{\hat{Z}} + \lambda \cdot d(X, \hat{X})$. Distortion $d(,)$ between X and $\hat{X}$ is measured by the expectation of mean squared error (MSE) in this study. λ is the factor that determines the rate-distortion trade-off. Bitrate $R_{\hat{Y}}$ and $R_{\hat{Z}}$ is estimated using the expected entropy of latent and hyperprior features. A fully factorized density model [1] is used to model the bit rate of $\hat{Z}$,

$$p_{\hat{Z}|\psi}(\hat{Z} \mid \psi) = \prod_i \left(p_{\hat{Z}_i|\psi^{(i)}}\left(\psi^{(i)}\right) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \right)(\hat{Z}_i) \tag{3}$$

where the vectors $\psi^{(i)}$ represent the parameters of each univariate distribution $p_{\hat{Z}^i|\psi^{(i)}}$ (all these parameters are collectively denoted as ψ) and $\mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right)$ means uniform distribution ranging from $-\frac{1}{2}$ to $\frac{1}{2}$. Conditioned on $\hat{Z}$, a conditional Laplacian distribution is used to estimate the probability density function (p.d.f.) of $\hat{Y}$,

$$p_{\hat{Y}|\hat{Z}}(\hat{Y} \mid \hat{Z}) = \prod_i \left(\mathcal{L}(\mu_i, \sigma_i) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \right)(\hat{Y}_i) \tag{4}$$

where the estimated mean and location parameters (μ_i, σ_i) of each element $\hat{Y}_i$ are derived from $\hat{Z}$. Finally, the bit rates of $\hat{Y}$ and $\hat{Z}$ are estimated using

$$R_{\hat{Y}} = -\sum_i \log_2 \left(p_{\hat{Y}_i|\hat{Z}_i}(\hat{Y}_i \mid \hat{Z}_i) \right), \tag{5}$$

$$R_{\hat{Z}} = -\sum_i \log_2 \left(p_{\hat{Z}_i|\psi^{(i)}}\left(\hat{Z}_i \mid \psi^{(i)}\right) \right). \tag{6}$$

3.2. Non-Local Attention Module

The implementation of typical non-local block (NLB) [20] in point cloud is shown in Figure 2a. An input feature map $M \in \mathcal{R}^{N \times C}$ is output from the previous layer, where N and C indicate the point number and channel number. Three 1×1 convolution W_ϕ, W_θ and W_γ are used to transform X to different features $\phi \in \mathcal{R}^{N \times \hat{C}}$, $\theta \in \mathcal{R}^{N \times \hat{C}}$ and $\gamma \in \mathcal{R}^{N \times \hat{C}}$ as

$$\phi = W_\phi(M), \quad \theta = W_\theta(M), \quad \gamma = W_\gamma(M), \tag{7}$$

where $\hat{C}$ is the channel number of the new embeddings. Then, the similarity matrix $V \in \mathcal{R}^{N \times N}$ is calculated by a matrix multiplication as

$$V = \phi \times \theta^{\mathrm{T}}. \tag{8}$$

Afterward, the softmax normalization is applied to V to get a similarity matrix as

$$\vec{V} = softmax(V). \tag{9}$$

The output of the attention layer is obtained by matrix multiplication as

$$O = \vec{V} \times \gamma, \tag{10}$$

where $O \in \mathcal{R}^{N \times \hat{C}}$. By referring to the design of the non-local block, the final output is given by

$$\tilde{M} = W_o\left(O^{\mathrm{T}}\right) + M, \tag{11}$$

where W_o, also implemented by a 1×1 convolution, acts as a weighting parameter to adjust the importance of the non-local operation with respect to the original input M and moreover, recovers the channel dimension from $\hat{C}$ to C. Considering global information, the non-local block effectively and adaptively diverts attention to the most important regions of an image in spatial dimension.

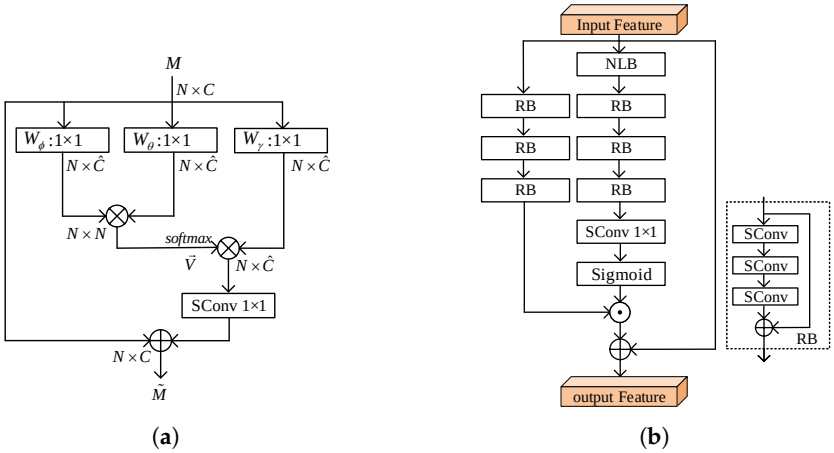


Figure 2. (**a**) non-local block. $\oplus$ is the add operation and $\otimes$ is the matrix multiplication. (**b**) non-local attention module. $\odot$ means element-wise product.

Inspired by the successful applications in computer vision of channel and spatial attention mechanism [24–27], we use the NLAM to generate feature maps in both dimensions. As detailed in Figure 2b, the NLAM unit includes the main branch, mask branch and connection branch. The main branch still processes features, and can be implemented by any state-of-the-art structure such as a residual or inception block. In this work, we stack three sparse convolutional residual blocks. The mask branch focuses on non-local operation to learn a mask of the same size that weights output features from the main branch, which includes the non-local block, followed three residual blocks (RBs), one 1×1

convolution ($\text{Conv}_{1\times1}$) and non-linear sigmoid activation. The sigmoid layer normalizes the output to [0, 1] and then the overall attention mask $\hat{M}$ can be written as

$$\hat{M} = \text{Sigmoid}(\text{Conv}_{1\times1}(\text{RBs}(\text{NLB}(X)))). \tag{12}$$

The third skip connection is used for faster convergence [28]. Then the final output of NLAM can be represented as

$$\tilde{X} = \left(\text{RBs}(X) \odot \hat{M}\right) + X. \tag{13}$$

where $\odot$ means element-wise product. The NLAM can provide fine-grained masks to distinguish the importance of latent features that will be further down sampled or compressed. Figure 3 visualizes the input point cloud *Bird* and its corresponding mask generated by the first NLAM. As shown in Figure 3e–h, the closer the color is to purplish red, the closer the weight is to 1, and the closer the color is to green, the closer the weight is to 0. The textured areas such as head and upper surface of the wing are allocated higher weights to retain the original features as shown in Figure 3b,c,f,g. The relatively plain areas such as belly and under surface of the wing are allocated lower weights to weaken their influence as shown in Figure 3d,h.

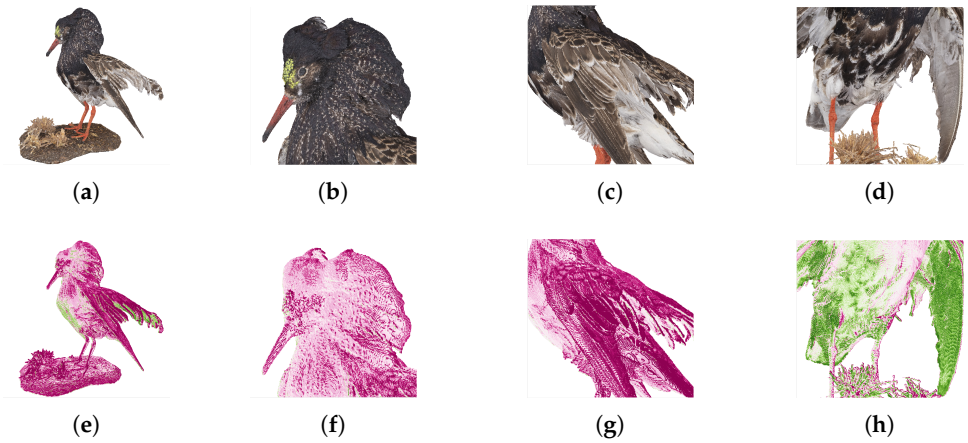


Figure 3. Visualization of input point and corresponding attention mask generated by the first NLAM. (**a**) the input point cloud *Bird*. (**b**) head. (**c**) upper surface of the wing. (**d**) belly and under surface of the wing. (**e**) the mask of *Bird*. (**f**) the mask of head. (**g**) the mask of upper surface of the wing. (**h**) the mask of belly and under surface of the wing.

3.3. Variable Rate Autoencoder Network

Variable rate is achieved by modulating the latent features of different layers in the encoder and the decoder, which means multiplying the feature map and the modulating/demodulating vector in a channel-wise production manner. Figure 4 shows the form of two-layer modulation, and in practice, the output of each convolution layer or attention module is modulated. Given a feature map X_i of channel i in the encoder, and the output map X_i' can be calculated as:

$$X_i' = m_i(\lambda)X_i, \tag{14}$$

where $m_i(\lambda)$ is the i-th element of the modulating vector $\mathbf{m}(\lambda)$. $\mathbf{m}(\lambda)$ depend on Lagrange multiplier λ by

$$\mathbf{m}(\lambda) = \exp(W_2\,\text{ReLU}(W_1\lambda)), \tag{15}$$

where W_1 and W_2 are weights matrices of two fully-connected layers. ReLU and exp are nonlinear activation and element-wise exponential function. The exponential nonlinearity guarantees positive outputs. The modulated feature maps in the decoder involve a similar calculation.

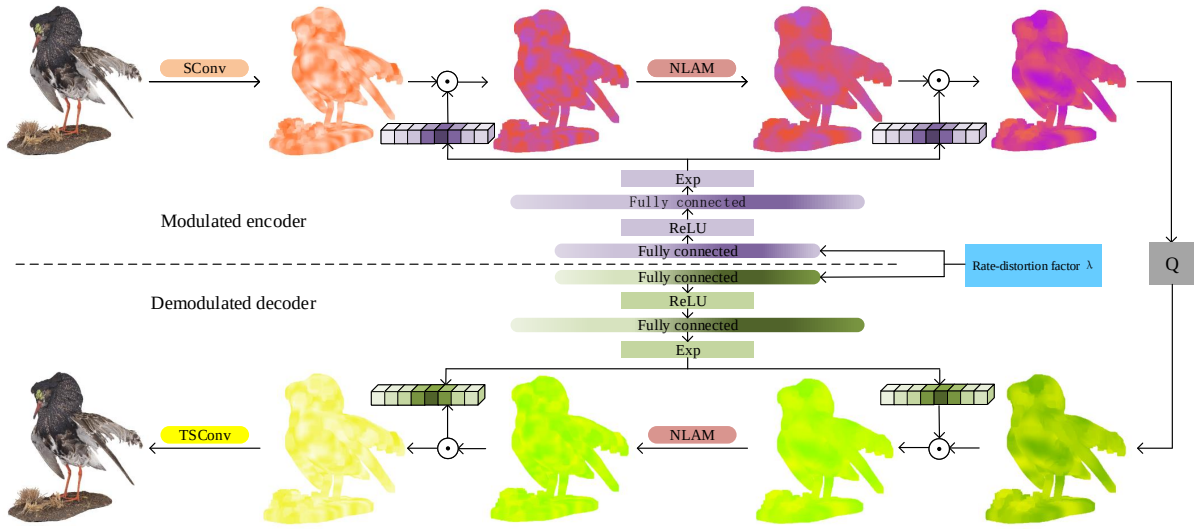


Figure 4. Variable rate autoencoder network with parameters shared modulation network . $\odot$ means channel-wise product.

Let the SConv layers and NLAMs collectively have learnable parameters θ, and modulating network has learnable parameters ϑ, and the decoding part has parameters ϕ and φ. As a result, the overall loss function becomes

$$\underset{\theta,\phi,\vartheta,\varphi}{\operatorname{argmin}} \sum_{\lambda\in\Lambda} [R(\hat{Y};\theta,\phi,\vartheta,\varphi,\lambda) + \lambda D(X,\hat{X};\theta,\phi,\vartheta,\varphi,\lambda)], \tag{16}$$

where Λ is a pre-defined finite set of λ. $R()$ is the estimated bitrate of latent features and $D()$ is the distortion between input point cloud attribute and reconstructed point cloud attribute.

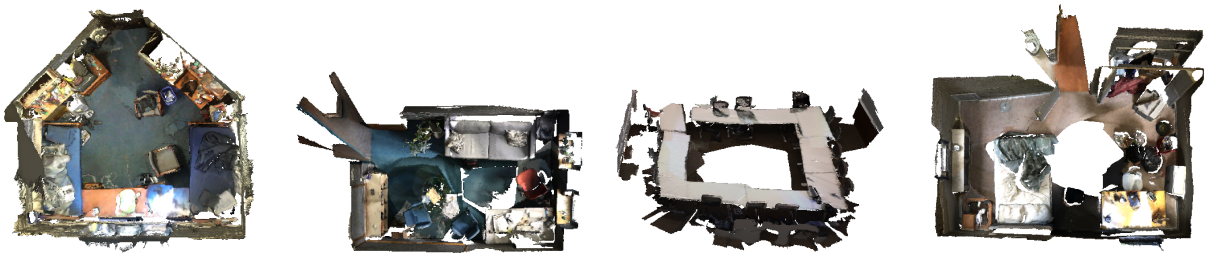
Although, for features of different layers, the parameters of the modulating network are shared, this mechanism allows multi-level and different modulation in case of more complicated networks in the future. In general, this architecture achieves effective joint training of both autoencoder and modulating network, therefore avoiding training and deploying multiple networks.

4. Experimental Results

This section describes the details of the experiments, including the train dataset, the test dataset, the anchors, and the hyper parameters for the model training. Moreover, objective quality and subjective quality comparison experiments have been performed that demonstrate the superiority of the proposed method.

4.1. Training Dataset

The publicly available ScanNet [29] is a large-scale dataset with dense point cloud surface reconstructions from the indoor scenario. It consists of 1513 scans and each scan contains tens of thousands to hundreds of thousands of colored points. In our experiment, each point cloud is quantified to 1 cm. Figure 5 shows some examples of the training dataset.



(**a**) (**b**) (**c**) (**d**)

Figure 5. Examples from ScanNet. (**a**) scene-0233 (**b**) scene-0262 (**c**) scene-0483 (**d**) scene-0699.

4.2. Training Details

We set λ to 0.01, 0.02, 0.05, 0.09 and 0.14 to obtain the model with different bitrates. The model is trained on the dataset for 100 epochs, with the learning rate decreased from 2×10^{-4} to 1×10^{-5}. The batch size is set to 8 and the optimizer is Adam [30] with parameter β_1 and parameter β_2 set to 0.9 and 0.999, respectively. The GPU device is Nvidia GeForce RTX 3090.

4.3. Rate-Distortion Efficiency

In a lossy compression problem, one must trade off two competing costs: the bitrate of compressed representation and the distortion between the reconstructed data and input data, which is also known as rate-distortion trade-off. Following the convention, we evaluate the proposed method by comparing the rate-distortion performance with different methods, including traditional and learning-based methods.

4.3.1. Test Dataset and Anchors

To demonstrate the performance of the proposed method, 8i Voxelized Full Bodies [15] are selected as test dataset. As shown in Figure 6, the geometry and attribute characteristics are completely different from the training dataset, which demonstrates the model generalization. We compare with two learning-based methods Deep-PCAC and SparsePCAC, and two different G-PCC versions: TMC13v6 and the latest and state-of-the-art TMC13v14.

(**a**) (**b**) (**c**) (**d**)

Figure 6. Test dataset. (**a**) longdress. (**b**) redandblack. (**c**) soldier. (**d**) loot.

4.3.2. Objective Quality Comparison

We follow the common practice to measure the bit rates (i.e., bits per points, bpp) and distortion in terms of Peak Signal-to-Noise Ratio (PSNR) in dB of the Y channel, which are computed using MPEG PCC pc error tools. As the bit rates generated by various algorithms are different, the BD-Rate (in percentage) is used to measure the overall R-D performance.

Figure 7 shows the R-D curves, and Table 1 shows BD-rate gains. The proposed method outperforms TMC13v6 by a large margin, having 31.5% BD-rate reduction. However, the proposed method still has performance losses compared with the state-of-the-art

algorithms TMC13v14. The results show that our scheme has 29.3% performance gap. For learning-based method SparsePCAC, the proposed method achieves 6.3% BD-rate reduction. For Deep-PCAC, the test sample "soldier" is used for comparison since the other three point clouds are used for training by Deep-PCAC. The proposed method achieves 50.7% BD-rate reduction.

Table 1. BD-rate(%) comparisons of the proposed method with TMC13 and PCAC.

	TMC13v6	TMC13v14	SparsePCAC	Deep-PCAC
	BD-Rate (%)	**BD-Rate (%)**	**BD-Rate (%)**	**BD-Rate (%)**
longdress	−33.2	20.1	1.6	-
solider	−39.1	23.8	−10.8	−50.7
redandblack	−38.5	5.8	−5.7	-
loot	−15.1	67.5	−10.2	-
average	−31.5	29.3	−6.3	−50.7

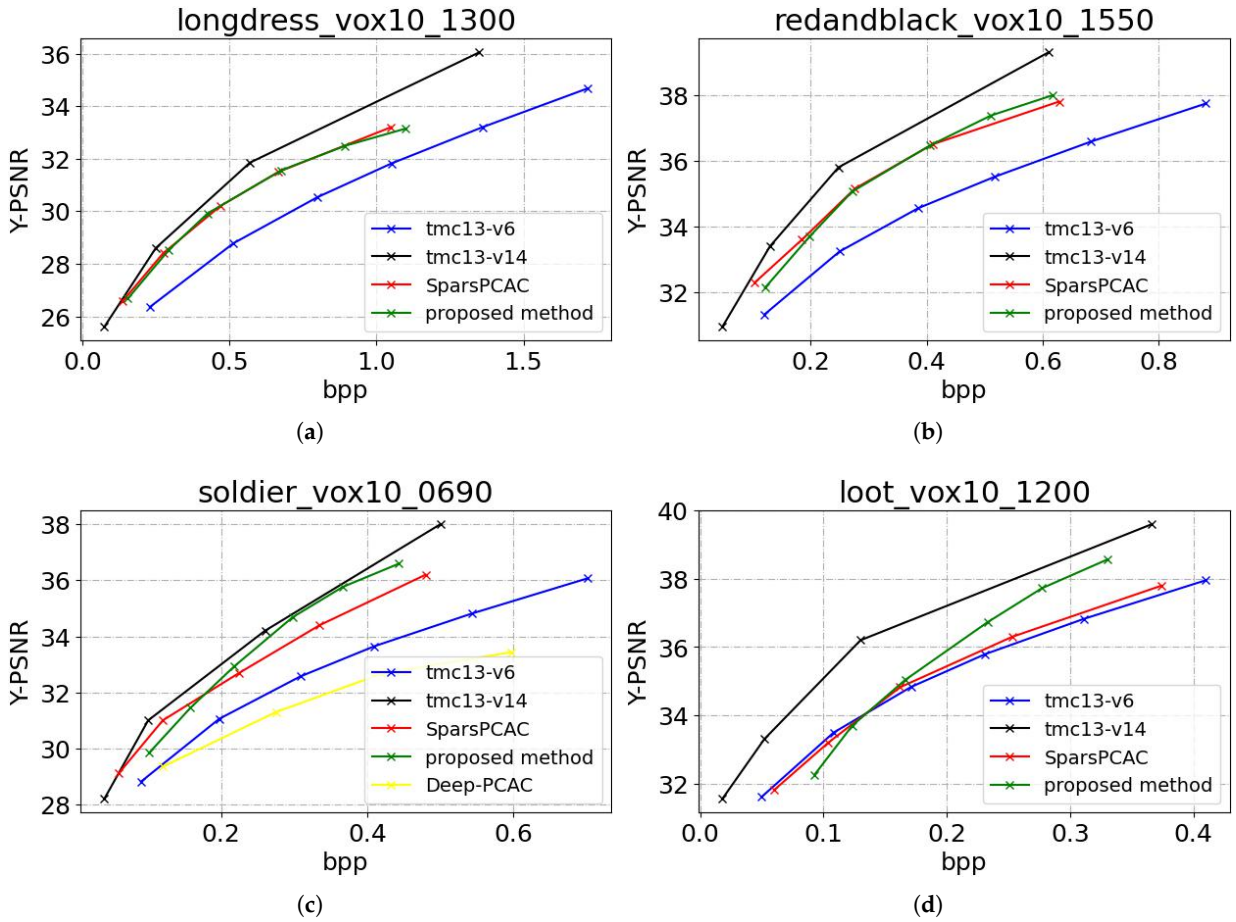


Figure 7. Performance comparison. (**a**) longdress. (**b**) redandblack. (**c**) soldier. (**d**) loot.

4.3.3. Subjective Quality Comparison

To show the benefits of the proposed framework in terms of subjective quality, we visualize the reconstructed point clouds of color attributes generated by different PCAC methods in Figure 8. For a fair comparison, we obtained the visual results of all methods at approximately equal bit rates. We can see that there are apparent blurry artifacts and blocky artifacts around the longdress's eyes and londress's fingers for TMC13v6 and TMC13v14. In general, our method achieves the best subjective quality by having smooth texture on flat regions and retaining more clear edges in sharp regions.

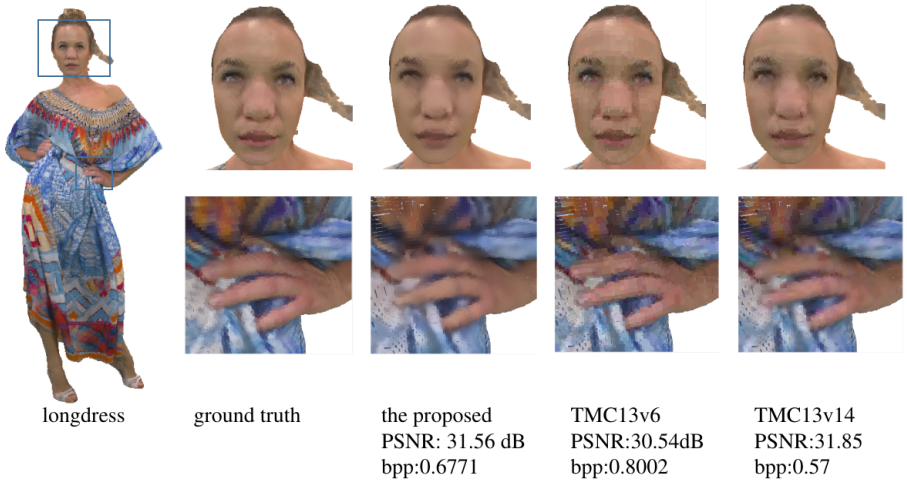


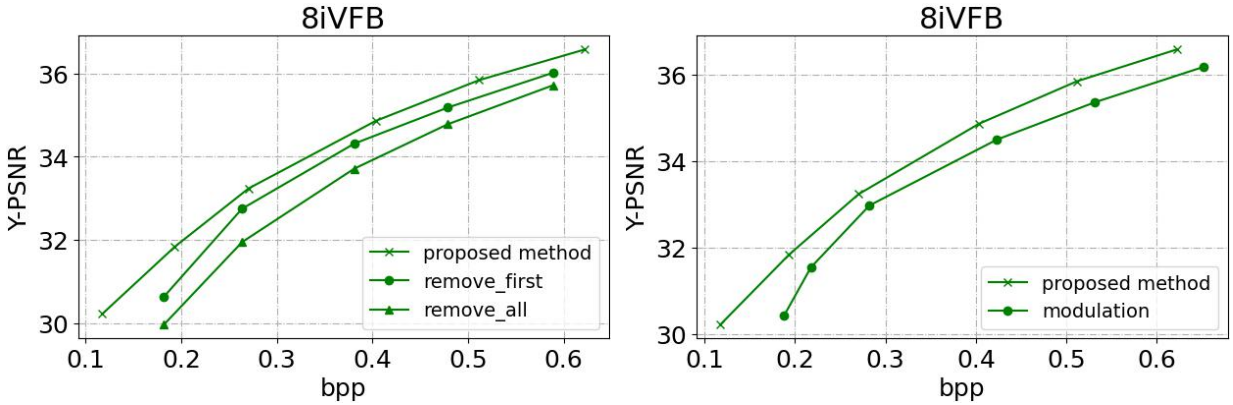
Figure 8. Subjective Evaluation. Visual comparison among TMC13v14, Ours, TMC13v6.

4.4. Ablation Studies

In this section, we present some ablation studies to further analyze the proposed framework in the following aspects to better understand the capability of our system in practice.

4.4.1. Influence of NLAM

To further make clear the contribution of the introduced NLAM, we set different configurations of network in terms of NLAM. The internal mask branch of NLAM pairs is gradually removed and the network is retrained for performance evaluation. Other experimental conditions remain unchanged and the result is shown in Figure 9a.



(**a**) (**b**)

Figure 9. Ablation studies. (**a**) influence of modulation network. (**b**) influence of NLAM.

Removing the mask branches of the NLAM pair in the shallow layer in the main encoder-decoders (referred to as "remove_first") yields a PSNR drop of about 0.3 dB compared to the orginal setting at the same bit rate. The drop is further enlarged when all NLAM pairs' mask branches are removed (a.k.a., "remove_all"), resulting in a network without any non-local characteristics explorations.

4.4.2. Influence of Variable Rates Modulation

A variable rates compression model is required for practical application, without retraining the model for individual rates. Since the original feature maps are changed by the modulation network in a nonlinear way, the data distribution may change, disturbing the entropy modeling. Figure 9b shows the comparison result of the proposed method

and the one adding modulation (referred to as "modulation"). As we can see in Table 2, although there is some performance loss, the number of parameters of the model has been significantly reduced, which is critical for applications.

Table 2. Average BD-rate over TMC13v6 and the number of parameters.

	BD-Rate	The Number of Parameters
original network	−31.5%	43.31 M
modulated network	−24.1%	8.09 M

5. Conclusions

In this paper, we propose an end-to-end learned point cloud attribute compression method with non-local attention optimization and a modulation network achieving variable rate compression. The proposed method achieves state-of-the-art performance compared with other existing learning-based methods, in terms of coding performance measured by BD-rate. When compared with traditional compression methods, the proposed method achieves a large improvement over TMC13v6 and further narrow the gap between the state-of-the-art TMC13v14. Specifically, for the BD-rate, the proposed method outperforms TMC13v6, Deep-PCAC and SparsePCAC by 31.5%, 50.7% and 6.3% reduction, respectively, and reduces the gap with TMC13v14 to 29.3%. For practical applications, the variable rate compression model is achieved to overcome the limitation of storing one network per bitrate. In future work, we would like to utilize the geometry information of the point cloud to extract the latent features of the attribute more reasonably and further optimize the corresponding distribution model.

Author Contributions: Conceptualization, X.H., S.Z. and F.Y.; methodology, X.H., S.Z. and F.Y.; software, X.H.; investigation, X.H. and S.Z.; writing, X.H. and S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (62171353, 62101409 and 61801364).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset used in this study can be found at http://www.scan-net.org/ (accessed on 15 March 2022). The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Ballé, J.; Laparra, V.; Simoncelli, E.P. End-to-end optimized image compression. *arXiv* **2016**, arXiv:1611.01704.
2. Ballé, J.; Minnen, D.; Singh, S.; Hwang, S.J.; Johnston, N. Variational image compression with a scale hyperprior. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, 30 April–3 May 2018.
3. Minnen, D.; Ballé, J.; Toderici, G. Joint Autoregressive and Hierarchical Priors for Learned Image Compression. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018; Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; pp. 10794–10803.
4. Agustsson, E.; Minnen, D.; Johnston, N.; Balle, J.; Hwang, S.J.; Toderici, G. Scale-Space Flow for End-to-End Optimized Video Compression. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 8500–8509.
5. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.

6. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; Volume 30.
7. Sheng, X.; Li, L.; Liu, D.; Xiong, Z.; Li, Z.; Wu, F. Deep-PCAC: An End-to-End Deep Lossy Compression Framework for Point Cloud Attributes. *IEEE Trans. Multimed.* **2022**, *24*, 2617–2632. [CrossRef]
8. Alexiou, E.; Tung, K.; Ebrahimi, T. Towards neural network approaches for point cloud compression. In Proceedings of the Applications of Digital Image Processing XLIII, Online, 24 August–4 September 2020; Tescher, A.G., Ebrahimi, T., Eds.; International Society for Optics and Photonics; SPIE: Bellingham, WA, USA, 2020; Volume 11510, pp. 18–37. [CrossRef]
9. Wang, J.; Wei, H.; Zakharchenko, V. *ISO/IEC JTC 1/SC 29/WG 7 m59037*; Point Cloud Attribute Compression Using Sparse Tensor-Representation; MPEG: Geneva, Switzerland, 2022.
10. Quach, M.; Valenzise, G.; Dufaux, F. Folding-Based Compression Of Point Cloud Attributes. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Negombo, Sri Lanka, 6–8 March 2020; pp. 3309–3313. [CrossRef]
11. Choy, C.; Gwak, J.; Savarese, S. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
12. Chen, T.; Ma, Z. Variable Bitrate Image Compression with Quality Scaling Factors. In Proceedings of the ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2163–2167. [CrossRef]
13. Yang, F.; Herranz, L.; Weijer, J.v.d.; Guitián, J.A.I.; López, A.M.; Mozerov, M.G. Variable Rate Deep Image Compression With Modulated Autoencoder. *IEEE Signal Process. Lett.* **2020**, *27*, 331–335. [CrossRef]
14. Choi, Y.; El-Khamy, M.; Lee, J. Variable Rate Deep Image Compression With a Conditional Autoencoder. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
15. d'Eon, E.; Harrison, B.; Myers, T.; Chou, P.A. *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) Input Document WG11M40059/WG1M74006*; 8i Voxelized Full Bodies-a Voxelized Point Cloud Dataset; MPEG: Geneva, Switzerland, 2017;
16. Zhang, C.; Florêncio, D.; Loop, C. Point cloud attribute compression with graph transform. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 2066–2070. [CrossRef]
17. de Queiroz, R.L.; Chou, P.A. Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform. *IEEE Trans. Image Process.* **2016**, *25*, 3947–3956. [CrossRef] [PubMed]
18. WG7. *ISO/IEC JTC 1/SC 29/WG 7 N0099*; G-PCC Codec Description v11; MPEG: Geneva, Switzerland, 2021.
19. Galassi, A.; Lippi, M.; Torroni, P. Attention in Natural Language Processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4291–4308. [CrossRef] [PubMed]
20. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-Local Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
21. Cheng, Z.; Sun, H.; Takeuchi, M.; Katto, J. Learned Image Compression With Discretized Gaussian Mixture Likelihoods and Attention Modules. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
22. Xue, Y.; Su, J. Attention Based Image Compression Post-Processing Convlutional Neural Network. In Proceedings of the CVPR Workshops, Long Beach, CA, USA, 15–20 June 2019.
23. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
24. Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; Chua, T.S. SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
25. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual Attention Network for Image Classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
26. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
27. Park, J.; Woo, S.; Lee, J.Y.; Kweon, I.S. Bam: Bottleneck attention module. *arXiv* **2018**, arXiv:1807.06514.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
29. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In Proceedings of the Proc. Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE: Piscataway, NJ, USA, 2017.
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.