EURASIP Journal on Image
and Video Processing

RESEARCH Open Access

Compressed point cloud classification with point-based edge sampling

Zhe Luo[1*], Wenjing Jia[1†] and Stuart Perry[1†]

[†]Wenjing Jia and Stuart Perry have contributed equally to this work.

*Correspondence: Zhe.Luo-1@student.uts.edu.au

[1] University of Technology Sydney, 15 Broadway, Sydney, NSW 2007, Australia

Abstract

3D point cloud data, as an immersive detailed data source, has been increasingly used in numerous applications. To deal with the computational and storage challenges of this data, it needs to be compressed before transmission, storage, and processing, especially in real-time systems. Instead of decoding the compressed data stream and subsequently conducting downstream tasks on the decompressed data, analyzing point clouds directly in their compressed domain has attracted great interest. In this paper, we dive into the realm of compressed point cloud classification (CPCC), aiming to achieve high point cloud classification accuracy in a bitrate-saving way by ensuring the bit stream contains a high degree of representative information of the point cloud. Edge information is one of the most important and representative attributes of the point cloud because it can display the outlines or main shapes. However, extracting edge points or information from point cloud models is challenging due to their irregularity and sparsity. To address this challenge, we adopt an advanced edge-sampling method that enhances existing state-of-the-art (SOTA) point cloud edge-sampling techniques based on attention mechanisms and consequently develop a novel CPCC method "CPCC-PES" that focuses on point cloud's edge information. The result obtained on the benchmark ModelNet40 dataset shows that our model has superior rate-accuracy trade-off performance than SOTA works. Specifically, our method achieves over 90% Top-1 Accuracy with a mere 0.08 bits-per-point (bpp), marking a remarkable over 96% reduction in BD-bitrate compared with specialized codecs. This means that our method only consumes 20% of the bitrate of other SOTA works while maintaining comparable accuracy. Furthermore, we propose a new evaluation metric named BD-Top-1 Accuracy to evaluate the trade-off performance between bitrate and Top-1 Accuracy for future CPCC research.

Keywords: Point cloud compression, Point cloud classification, Compressed domain, Edge sampling

1 Introduction

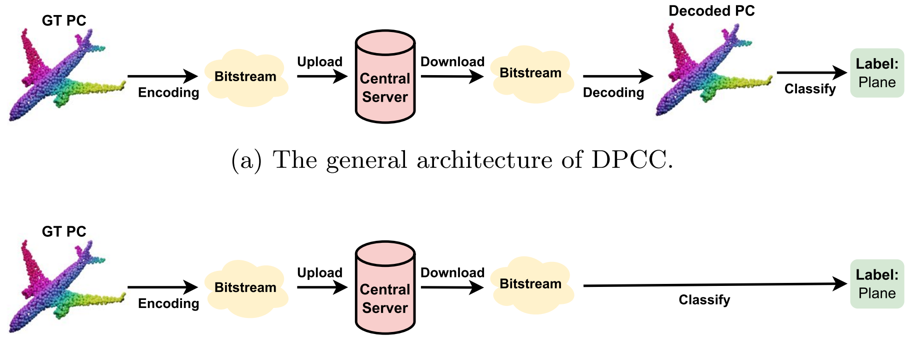
Point clouds are widely used in numerous applications, such as autonomous driving, augmented reality, virtual reality, robotics, smart cities, and environmental sensing. Point clouds encapsulate comprehensive perception information, including 3-dimensional (3D) geometry coordinates, color, geodesic distance, normals, and point cloud density. However, these rich details often result in massive file sizes, presenting formidable challenges in storage, transmission, processing, and computation. Usually, advanced deep learning techniques for downstream vision tasks have higher requirements on computational resources, which is not economically friendly.

Addressing these physical and economic limitations, one effective strategy is decompressed point cloud analysis (DPCA). Here, point clouds are analyzed and compressed into bit streams at the scanning or server-side terminal. These bit streams are then transmitted to the central computing hub which is equipped with powerful computational resources, where they are effectively decompressed into decoded point clouds. Then the computing hub conducts subsequent downstream vision tasks and dispatches the analyzed results back to the end devices to guide the decision-making of users. While this strategy preserves original point cloud details and enhances model efficacy, it requires not only high computational resources but also stable and fast transmission mediums such as cables and fiber optics. In addition, decoded point clouds often exhibit artifacts such as outliers or shrinkage, which compromises the performance of the downstream tasks compared to performing the task directly on the original point cloud.

In response to these challenges, our focus shifts toward compressed point cloud analysis (CPCA), which identifies a more feasible and economically friendly bitrate-saving solution for effective downstream tasks processing. As shown in Fig. 1a, instead of performing point cloud classification on the decoded point cloud in decompressed point cloud classification (DPCC), compressed point cloud classification (CPCC) conducts point cloud classification directly on the encoded bit stream. While existing approaches, such as Refs. [1, 2], have revealed the potential of CPCC, their classification performance still falls behind state-of-the-art (SOTA) point cloud classification results.

To enhance CPCC's efficacy, in this paper, we propose a novel compressed point cloud classification approach, abbreviated as CPCC-PES, featuring a point-based edge sampling module, which focuses on preserving point cloud edge information for improved classification.

Edge information, widely used in 2D image classification and segmentation, captures crucial representative semantic information of shape contour. Whereas, in the field of point cloud processing, even though many advanced techniques have been applied in storing representative semantic information for the point cloud in a learnable way, many of them have not considered the shape outlines and edge information as special



(a) The general architecture of DPCC.

(b) The general architecture of CPCC.

Fig. 1 The comparison of the general architectures of DPCC and CPCC

and representative features of the point cloud. Our aim is to extend the success of 2D edge-preserving processing into 3D point cloud domain. However, it is difficult to detect edges of point clouds efficiently because point clouds are often irregular, containing regions of different density and sparsity. Thus, in this paper, to achieve higher classification accuracy at a smaller bitrate, we design our model based on a novel point cloud edge sampling method.

In traditional 3D point cloud analysis, currently widely used mathematical sampling methods include farthest point sampling (FPS) [3], random point sampling [4], and grid sampling [5]. Integrated with deep learning techniques such as Convolutional Neural Networks and Transformers, these down-sampling methods can effectively capture the essential structural information of a point cloud for improving the performance of learning-based downstream tasks. To add more adaptability to different downstream tasks, some advanced learning-based down-sampling methods have been proposed, tailored to different downstream tasks, e.g., SampleNet [6], Skeleton-aware Down-sampling [7], DA-Net [8], and LightN [9].

Even though many advanced techniques have been applied utilizing the representative semantic information of point clouds in a learnable way, many of them have not considered the shape outlines and edge information as special and representative features of the point cloud. In this work, we propose a codec to utilize an attention-based edge-sampling method that focuses on salient outlines and the shape of the point cloud model to enrich the bit stream and the latent features with more boundary-related semantic information and shape outline details. Our results show that our model has achieved superior performance compared to the SOTA by selecting the sampled points in an adaptive way.

In short, we summarize our contributions as follows:

- We developed a new “CPCC-PES” model for CPCC by adopting an advanced edge-sampling method that focuses on the salient outlines and shape of the point cloud model to enrich the bit stream and the latent features with more boundary-related semantic and outline shape details.
- We designed an attention-based learnable edge-sampling approach by incorporating local attention mechanisms, demonstrating superiority over DPCC through benchmark dataset experiments.
- We conducted comprehensive experiments on a widely used benchmark dataset and achieved SOTA Bjøntegaard Delta (BD) Rate and Top-1 Accuracy performance using less bitrate to get higher Top-1 Accuracy simultaneously.

2 Related work

2.1 Point-based point cloud analysis

Since the introduction of PointNet [10] and PointNet++ [11], point cloud analysis (PCA) based on point-wise information has been widely used in many existing works. This end-to-end learning architecture allows models to directly utilize the point-based representative information of the point cloud without necessitating additional data preprocessing like voxelization. Inspired by this autoencoder architecture, several supervised and self-supervised learning PCAs have integrated multiple effective feature learning operations, such as point convolution and point attention, to enhance latent representations. For instance, PointConv [12] and KPConv [13] introduced point-wise convolution operators, in which points were convoluted with their neighbor points. A more advanced operator, PointConvFormer [14], builds an attention-based weight filter to select relatively semantically similar neighbor points for convolution.

In addition to convolution operations, attention mechanisms have attracted great attention for feature extraction due to their suitability for handling irregular point cloud data. Notable works in this realm include Point Transformer [15], which aggregates local neighbor attention, Point Cloud Transformer [16], which enlarges the receptive field into the whole input and utilizes global point cloud attention, Geo-former [17], which utilizes point cloud geodesic information to search and choose neighbor points with similar outlier semantic attributes and then perform local neighbor attention, Point4Transformer [18], which adds offsets between a local point and its neighbors to generate the convolution kernels to integrate local and global information, and Point Transformer v2 [5], which uses grouped vector attention to mitigate the high attention-related computational cost and generalization restriction problem with a stronger position embedding. In addition, graph-based methods analyze point clouds using a graph structure. For instance, in DGCNN [19], EdgeConv blocks update the neighbor information dynamically based on dynamic graphs.

2.2 Point cloud analysis in compressed and decompressed domains

As the demand for end-device usage increases and the development of learning-based point cloud compression continues to evolve, point cloud analysis in the compressed domain, i.e., CPCA, is emerging as an interesting topic. This approach enables the end devices to conduct the point cloud downstream tasks in a resource-efficient way. There are some existing CPCA methods focusing on different aspects of this domain. For instance, the deep learning-based compressed domain point cloud classification approach presented in Seleem et al. [20] introduced a bridge layer to connect the latent representation of the JPEG Pleno Point Cloud coding Verification Model [2] to a concise Point Grid Partial Classifier [21] to predict classification labels in a voxel-based way. Meanwhile, Ulhaq and Bajic in Ulhaq and Bajic [1] designed an end-to-end Learned Point Cloud Compression network for Classification (LPCCC) based on PointNet [10]. By adding two gain layers in the entropy bottleneck between the encoder and decoder, LPCCC [1] can increase the model's adaptability stability when coding the point cloud's latent geometry representation while maintaining high accuracy. It has been shown that LPCCC [1] performs better than some advanced DPCC methods that utilize SOTA codecs such as OctAttention [22], G-PCC [23], and IPDAE [24] to compress and decompress point clouds and then utilize PointNet [10] to classify the decompressed point cloud. Whereas in DPCA, Bojun et al. [25] built a gradient bridger to pass the gradient from a codec's decoded point cloud to a detection network using a point-matching method.

Despite the potential and advantages demonstrated by these advanced methods in achieving comparable or superior performance while conserving bitrate compared to traditional uncompressed point cloud analysis methods, there remain limitations and avenues for further enhancement. For example, in the compressed domain, existing methods have only focused on point cloud detection and classification problems, and yet there are other widely used applications waiting to be researched, such as point cloud segmentation or point cloud registration. Furthermore, the performance of existing CPCA methods can be further improved. For example, PointNet [10] employed in Ulhaq and Bajic [1] is a classic point cloud analysis method that can be improved by some other advanced modules and strategies, such as the attention mechanisms [26].

2.3 Point cloud down-sampling

Point cloud down-sampling is a crucial pre-processing technique, aimed at capturing or preserving the overall distribution or structure of the original point cloud using a smaller subset of points. It enables the learning-based PCA to manage irregular data in an efficient way. Generally, traditional down-sampling approaches are the mainstream approaches because they can be integrated with deep learning techniques seamlessly, offering a consistent input size to the subsequent encoder stages. For example, farthest point sampling (FPS) [3] has been widely used in various popular PointNet variants to preserve the general representative information and structure of the point cloud. These include PointNet++ [11], Point-MAE [27], PT [15], PCT [16], and Stratified Transformer-3D [28]. As another example, Grid-sampling is used in Point Transformer v2 [5] to support point neighbor pooling during point cloud segmentation.

In addition to these conventional methods, many learning-based task-oriented down-sampling methods have shown their great potential in PCA. For example, S-Net [29] utilizes the global representation of the point cloud to generate new point cloud geometry coordinates. Inspired by this work, SampleNet [6] extends the post-processing by employing projection actions to better predict the output points, while PST-NET [30] introduces attention to improve feature learning, and LightN [9] proposes a lightweight Transformer framework plug-in to boost efficiency. Although these frameworks can improve the performance of PCA by adding adaptive capability, they often fail to explore 3D object geometries [7] explicitly. Considering that semantic information is a significant representative feature of point clouds, some SOTA works have introduced semantic-oriented down-sampling strategies. For example, Skeleton-aware Down-sampling [7] utilizes the medial axis to establish the prior knowledge of the point cloud skeleton, enabling unsupervised skeleton-aware sampling.

Despite the benefits and successes of these popular down-sampling methods in point cloud feature learning, maintaining the main or important semantic structure information of the point clouds, they are not designed for capturing the point cloud outliers and edge information.

Edge information, widely and successfully applied in 2D image feature learning, is also one of the most significant representative sources of information for 3D point cloud models, considering their sparsity and irregularity. An effective edge-sampling method to gather the representative edge information is vital for improving the performance of CPCC because when the number of feature channels increases, the number of down-sampled points will decrease during feature encoding. These edge-focused down-sampled points can preserve the semantic information of the point cloud's representative outliers, thereby benefiting classification.

In our CPCC-PES approach, we utilize attention-based edge sampling to capture the edge information of the point clouds while selecting the corresponding embedded point cloud features to improve the classification performance while using a lower bitrate.

3 The proposed method

3.1 The overall architecture of CPCC-PES

As shown in Fig. 2, CPCC-PES consists of an encoder layer, an entropy bottleneck layer, and a decoder layer. The input point cloud is converted to high-dimension latent features through an embedding layer formed by Multilayer Perceptrons (MLPs). Then the Neighbor Attention Layer learns and transfers these latent features to the attention-based edge-sampling layer. This down-sampling layer selects the edge information and point cloud features based on local attention, which serves as an ideal normalized correlation map to measure their difference, providing a learnable adaptive approach for edge point and feature selection. Inspired by the attention-based point cloud edge sampling (APES) work in Wu et al. [31], the normalized correlation map is computed by the attention map between the center point and its neighbors to identify how much the features of surrounding points differ from the central point (see Eq. 1 in Sect. 3.3 for more details). A larger standard deviation in the normalized correlation map means a higher possibility that it is an edge point. This edge-sampling method mitigates the feature learning limitation of the encoder of LPCCC [1] and achieves a better trade-off between accuracy and bitrate. Moreover, in contrast to APES [31] and inspired by advanced learning-based point cloud methods such as D-PCC [32], we utilize three down-sample layers to better control bitrate accompanied with a down-sampling ratio of 0.5. After each down-sample layer, we can obtain the edge points of the point cloud. Then the corresponding representative edge features will be selected and learned based on these edge points through a Neighbor Attention Layer. These learned features will concatenate together and pass through an MLP layer and a pooling layer to generate the final input permutation-invariant feature vector. Finally, the vector will pass an entropy bottleneck layer which is added to two gain layers to get the coded bit stream to increase its adaptive capability and obtain better performance. In the decoder, the decoded feature vector will pass through an MLP layer to get the predicted label.

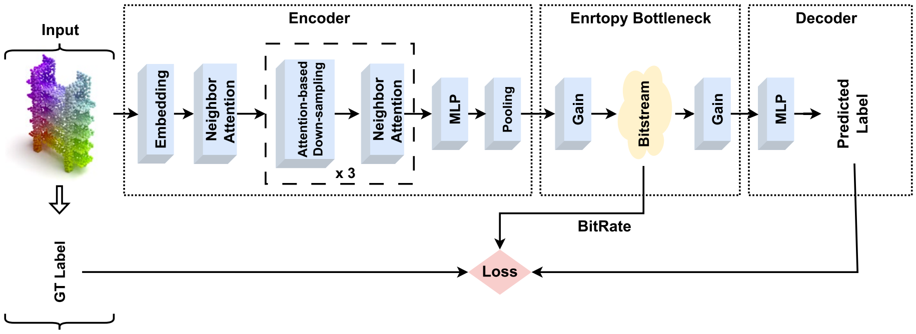


Fig. 2 The architecture of our CPCC-PES network

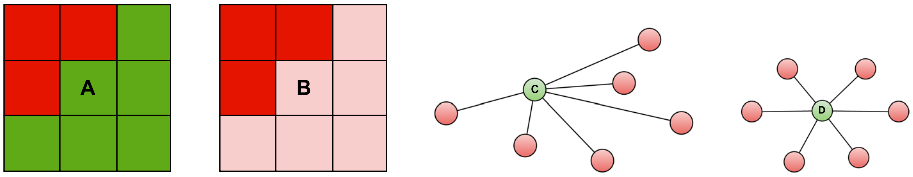


Fig. 3 Illustration of using standard deviation to select edge pixels (left) or edge points (right). We first calculate each normalized correlation map between the pixel/point and its neighbors, where the center pixel/point is self-contained as a neighbor. Then we compare the standard deviation values of edge pixels *A* and *B*, and points *C* and *D* to find out which one is an edge pixel or edge point

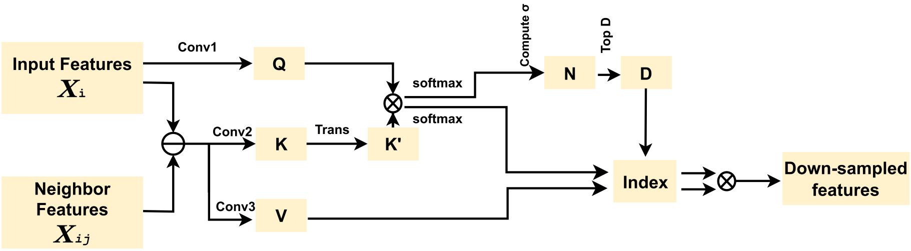


Fig. 4 The architecture of local-attention-based edge sampling

3.2 Attention-based edge sampling

The Canny edge detector [33] is widely used in edge detection for 2D images. When applying it, the intensity gradient of each pixel of the image, such as the strength of a color change, is computed and then compared with other pixels' gradients within a local patch area. The pixels which have larger gradient intensity difference are recognized to be the edge pixels, which can outline the edge details of the images.

In detail, we calculate a normalized correlation map and each pixel's local neighborhood standard deviation σ and then select the edge pixels which have higher σ values to differentiate them from other pixels.

For instance, as illustrated in Fig. 3, denoting the standard deviation as σ, the standard deviations of edge pixels A and B and points C and D can be represented as σ_A, σ_B, σ_C, and σ_D, respectively. Suppose σ_A is bigger than σ_B and σ_C is bigger than σ_D, then pixel A and point C will have a larger possibility of being an edge pixel or an edge point.

Specifically, to extend the edge detection to irregular point cloud areas, we use k-Nearest Neighbor (kNN) to define a local patch, and we also compute a normalized correlation map to find and differentiate the edge points within a point patch. Inspired by Wu et al. [31], the attention map is an ideal option to serve as the normalized correlation map between point features within each patch.

Assuming that the input point cloud set has n points, which can be represented as $\boldsymbol{P} = \{\rho_i\}_{i=1}^{n} \in \mathbb{R}^{n\times 3}$. For each point ρ_i, we find its neighbor point ρ_j by kNN to form a point patch. Thus, as shown in Fig. 4, within a patch, the input features for the center point ρ_i can be represented as X_i, the corresponding features of the neighbor point ρ_j are X_{ij}, and the feature difference between the center point ρ_i and its neighbor ρ_j can be represented as $X_{ij} - X_i$. Then X_i will be sent to $conv1$ to get the query Q. Meanwhile, $X_{ij} - X_i$ will be sent to $conv2$ and $conv3$, to obtain the key K and the value V, respectively. Following the design of attention mechanism [26], the attention map can be computed as QK^T. In this context, the local attention correlation map can be calculated as:

$$\mu(X_i, X_{ij}) = \psi(X_i)^T \varphi(X_{ij} - X_i). \tag{1}$$

Here, ψ and φ are fully connected layers where inputs are the center point features X_i and the feature difference between the neighbor point and the center point, $X_{ij} - X_i$, respectively. $\psi(X_i)$ is the query Q and $\varphi(X_{ij} - X_i)$ is the key K.

Because calculating the attention also involves *softmax* normalization and scaling with a factor $\sqrt{d}$, the final equation of the normalized correlation map $\boldsymbol{M}_i$ regarding point ρ_i can be written as:

$$\boldsymbol{M}_i = \text{softmax}\Big(\mu(X_i, X_{ij})/\sqrt{d})\Big). \tag{2}$$

After we have obtained the correlation map $\boldsymbol{M}_i$, the standard deviation σ_i of the map $\boldsymbol{M}_i$ can be calculated. We can then select the edge points' indexes and their corresponding features by selecting those points with a higher σ_i. For example, in Fig. 4, we obtain the down-sampled index by selecting D points from the N points based on the top σ_i values and then obtain the down-sampled features.

3.3 Neighbor attention aggregation

After each attention-based down-sampling layer, we can obtain the down-sampled point cloud edge points and their corresponding features. Considering that the down-sampling ratio is set to 0.5, the dimensions of the down-sampled point features will be 1024, 512, and 256 at each stage. To further increase the performance of CPCC-PES, we decided to incorporate an attention mechanism [26] here to learn and aggregate the obtained down-sampled point features. We make use of the attention mechanism from APES [31] as a stronger tool to capture the features that are focused on the edge information of the point cloud.

We first find the center point and its k corresponding neighbor points using the kNN and set k to 32. As shown in Fig. 5, the features of the center point are set as the input features and are passed through a *conv*1 layer to get Q, and the feature difference between the center point and its neigbors will pass through *conv*2 and *conv*3 separately to get K and V. Then the equation of the neighbor attention features $\boldsymbol{F}_N$ can be represented as follows:

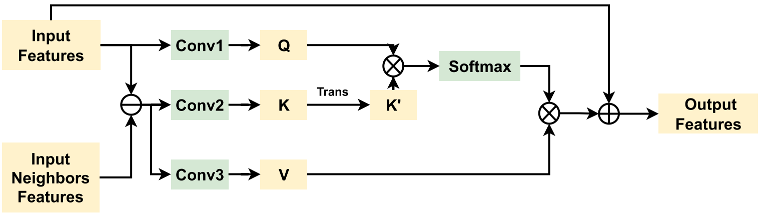


Fig. 5 Neighbor attention aggregation

$$\boldsymbol{F}_N = \mathrm{softmax}\Big((QK^T)/\sqrt{d} \Big) V, \tag{3}$$

where $\sqrt{d}$ is the scaling factor. At last, the output feature will be the concatenation between the neighbor attention features $\boldsymbol{F}_N$ and the input features.

3.4 Entropy bottleneck

Gain layer: After max-pooling and normalization in the encoder, the result vectors are composed of all small values, which is not friendly to the following entropy calculation and also the classifiers in the decoder. To increase the adaptability and stability of the entropy bottleneck, we follow the ideas from LPCCC [1] to add two gain layers. Each gain layer is defined to be formed with a trainable vector, and the elements in this trainable vector are initialized with integers. In our experiment, we set the initial values of each of the elements of this trainable vector to 10. Then these trainable vectors will be multiplied element-wise with the input feature vectors to amplify the small values and differences existing in the input.

Entropy model: The first gain layer will receive the learned feature vectors provided by the encoder and output an adaptive version of the feature vector. To further obtain the bit streams, we use integer rounding to quantify these vectors. During training, uniform quantization is simulated using additive uniform noise $\mu \in (-0.5, 0.5)$. Then the quantized vector is losslessly encoded using a fully factorized learned entropy model [34].

3.5 Decoder

To reduce the overall computational complexity, we follow the work of LPCCC [1] and design a lightweight decoder module. In our decoder, we use a lightweight MLP classification layer that turns the initial decoded bit stream into predicted labels.

3.6 Loss function

In the point cloud compression field, researchers use a trade-off between bitrate and distortion loss which is measured by Chamfer Distance to quantify the performance of the compression model. Similarly, in CPCC, a trade-off between bitrate and classification accuracy can also be considered.

Following the idea of information bottleneck (IB) [35], LPCCC [1] has given evidence that the below equation, as a trade-off between bitrate and accuracy, is analogous to the IB, and it is suitable to be the loss function in CPCC:

$$L = R + \lambda \cdot D(t, \hat{t}). \tag{4}$$

Here, L represents the total loss value, R represents the bitrate value calculated by the entropy model, λ represents the penalty weight, and $D(t, \hat{t})$ represents the cross-entropy loss between one-hot encoded ground truth label t and the softmax of the prediction label $\hat{t}$ obtained by the model. In our approach, we also utilize this method to calculate the total loss.

4 Experimental results and discussion

4.1 Dataset and implementation details

Following the main trend of point cloud classification, we use ModelNet40 [36] as our test dataset, which contains 12,311 manufactured 3D CAD models in 40 common object categories. The dataset is split into training with 9843 models and testing with 2468 models. For each model, we uniformly sample points from the mesh surface and normalize them to the unit sphere. The input only contains point cloud coordinate information and no data augmentation methods are used.

Our model was trained and tested on a computer with an i9-13900K 3.61GHz CPU, 24 GB RAM, and an NVIDIA Ge-Force RTX 4090 GPU. During training, we set the penalty weight λ in the loss function (see Eq. 4), which ranges from 20 to 16,000, the initial learning rate was 0.01, batch size of 8, and Adam optimization.

4.2 Performance comparison and evaluation

As a common evaluation metric, Top-1 Accuracy has been widely used in many CPCC works, such as VM-CPCC [20] and LPCCC [1]. In our experiment comparison, because VM-CPCC is a voxel-based CPCC method, we compare our results with point-based LPCCC [1] in a compressed way and other SOTA works including G-PCC TMC13 [37], OctAttention [22], and IPDAE [24] in a decompressed way, as shown in Table 1. In addition to the conventional metric BD-Rates [38], we also propose an evaluation metric called BD-Top-1 to evaluate the different models' power in achieving less bitrate for higher accuracy.

Top-1 classification results on ModelNet40: To demonstrate the efficiency of the proposed CPCC-PES, we compare our method with SOTA DPCC methods. In addition, to highlight and compare the differences in bitrate expenses across various CPCC and DPCC techniques, we compare CPCC-PES against the baseline method PointNet [10], which is commonly employed as a baseline in many DPCC tasks, such as the transformer-based APES [31] and LPCCC [1]. We apply these methods to the same point cloud dataset and utilize the same input for encoding and decoding. Then following the practice in LPCCC [1], we employ the PointNet [10] classifier to obtain the final classification result. Figure 6 illustrates the comparative results.

From Fig. 6, we can see that our codec has superior performance compared with the rest of the SOTA codecs, demonstrating that our codec can achieve higher classification at lower bitrates. It is worth noting that PointNet [10] proposed input geometric transformations to align point clouds for better classification performance and reported their results with and without the affine transformations. Since our compression model was trained without the input spatial alignment, the baseline without geometric transformations offers a fairer comparison.

This comparison highlights the differences in bitrate costs among various CPCC and DPCC methods. In particular, considering the baseline PointNet [10]'s Top-1 Accuracy is 89.2%, we achieve the baseline by only consuming about 0.06 bpp (bits per point) and 0.07 bpp. We can also achieve 90.8% Top-1 Accuracy with only 0.2 bpp cost, while LPCCC [1] needs to spend 0.45 bpp to obtain a Top-1 Accuracy of 88.5%.

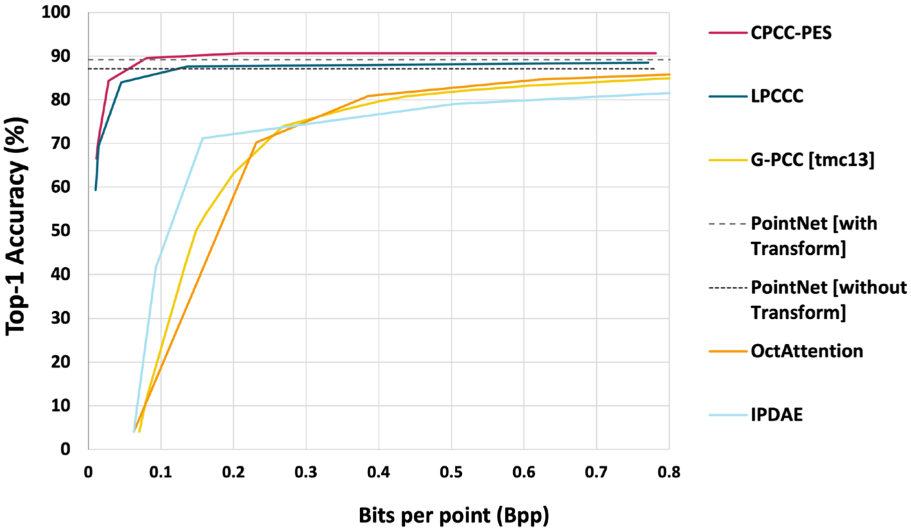


Fig. 6 Top-1 classification results on ModelNet40 at different Bpp rates

Table 1 Comparison of BD-Rate and BD-Top-1 accuracy with SOTA codecs

Compared with CPCC method	BD-Rate (%)	BD-Top-1 Accuracy
CPCC-PES vs LPCCC [1]	− 24.24	3.02
Compared with DPCC methods	**BD-Rate (%)**	**BD-Top-1 Accuracy**
CPCC-PES vs IPDAE [24]	− 96.12	22.98
CPCC-PES vs OctAttention [22]	− 94.54	30.2
CPCC-PES vs G-PCC [37]	− 95.2	15.2

BD-Rate and BD-Top-1 Accuracy: The video standardization community has used Chamfer Distance and BD-PSNR [39, 40] measurements for many years as a method for evaluating the performance of new codecs and video coding tools. In particular, BD-PSNR can measure the average rate difference between two PSNR rate-distortion curves and can be calculated as:

$$\text{BD-PSNR} = \frac{1}{R_1 - R_2} \int_{R_1}^{R_2} [\text{PSNR}_1(R) - \text{PSNR}_2(R)]\, dR \tag{5}$$

where R_1 and R_2 are the distortion limits, and $\text{PSNR}_1(R)$ and $\text{PSNR}_2(R)$ are the PSNR values of two codecs at bitrate R.

Following the metric of BD-PSNR, we propose the BD-Top-1 Accuracy metric, denoted as BD-Top-1Accu, to evaluate the performance of different CPCC codecs in terms of their bitrate efficiency while achieving high Top-1 Accuracy. BD-Top-1 Accuracy quantitatively provides a comprehensive insight into the codec's ability to maintain high point cloud classification accuracy at minimized bitrate levels. Similar to the design of BD-PSNR, our BD-Top-1 Accuracy metric is defined as:

$$\text{BD-Top-1Accu} = \frac{1}{R_1 - R_2}\int_{R_1}^{R_2}\left[\text{Top-1Accu}_1(R) - \text{Top-1Accu}_2(R)\right] dR \qquad (6)$$

where BD-Top-1 Accu is the BD-Top-1 Accuracy, and the Top-1 Accu_1(R), Top-1 Accu_2(R) are the Top-1 Accuracy values of two codecs at bitrate *R*.

Table 1 shows the comparison of BD metrics with the maximum attainable accuracy with SOTA methods. Following [38], in our evaluation metrics, BD-Rate measures the relative bitrate value difference between two CPCC performances. A CPCC method with a lower BD-Rate will spend less bitrate while achieving the same point cloud classification accuracy. In general, our model can save over 24% bitrate compared to the SOTA methods. We also save over 95% bitrate to achieve the same accuracy performance when comparing our method with the DPCC methods.

4.3 Ablation studies

In this section, we investigate the potential of our CPCC-PES approach for smaller datasets in CPCC tasks by comparing the results with the SOTA work.

Performance of CPCC-PES with small numbers of points: Typically, each point cloud model contains 2048 points, randomly sampled from the ModelNet40 [36] dataset. To assess the adaptability of the proposed CPCC-PES to smaller point cloud models, we reduce the input point cloud size using a random sampling technique to 1024, 512, and 256 points.

Table 2 compares the Max Top-1 Accuracy results of our method and those of the SOTA LPCCC [1] work under different numbers of input points. Max Top-1 Accuracy refers to the maximum Top-1 Accuracy that the method can achieve in point cloud classification without considering the bitrate consumption.

The result in the table shows that our method consistently outperforms LPCCC [1] in Max Top-1 Accuracy even with reduced point cloud size. This shows that our CPCC method is robust to different sizes of point clouds. Specifically, we achieve about 2% higher Max Top-1 Accuracy compared to the SOTA work LPCCC [1].

4.4 Edge-sampling visualization

Figure 7 below visualizes the local-attention-based edge-sampling point cloud models obtained from ModelNet40 [36]. From top to bottom, the point clouds displayed represent *television*, *plane*, *desk*, and *bed*. From left to right, each model contains 2048, 1024, 512, and 256 points, respectively. It can be seen from the figure that, using the

Table 2 Comparison of Max Top-1 Accuracy with the SOTA work LPCCC [1] concerning various numbers of input points

Number of input points	CPCC-PES Max Top-1 Accuracy (%)	LPCCC [1] Max Top-1 Accuracy (%)
1024	**90.8**	88.5
512	**90.35**	88.0
256	**90.02**	87.6

The best performance in each experiment is indicated in bold

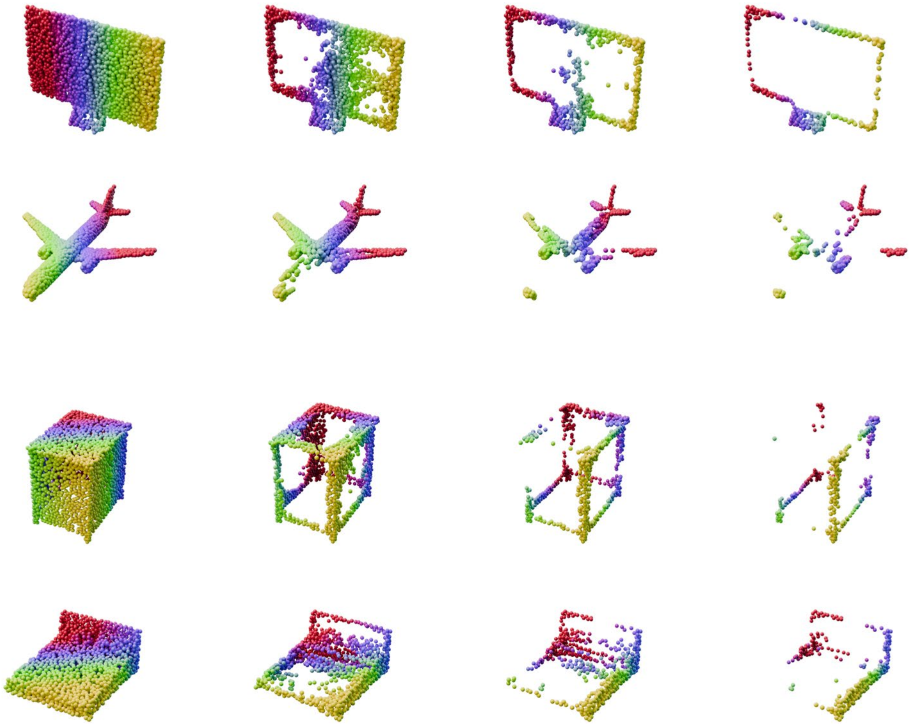


Fig. 7 A visualization of edge point cloud sampling. From top to bottom, the point clouds are *television*, *plane*, *desk*, and *bed*. From left to right, each point cloud model contains 2048, 1024, 512, and 256 points, respectively

local-attention-based attention map, we can get the primary structural-outlier edge information of the point cloud.

5 Conclusion

In this paper, we have proposed a new approach for point cloud classification in compressed domain by leveraging attention-based point cloud edge sampling. By extending the success of edge detection to point cloud in an effective way, our approach enhances the classification accuracy and performance at reduced bitrate. Our experiments demonstrate that our method shows a BD rate reduction of over 24% compared with SOTA methods. For future work, there are several promising directions for further exploration. These include exploring end-to-end compressed point cloud segmentation, integrating neural network compression and compressed point cloud downstream codecs, and developing a more efficient classifier with reduced computational costs.

Acknowledgements
We thank Yiran Guo for their invaluable support in data visualization.

Data availability
The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing intetests
The authors declare that they have no competing interests.

Received: 16 February 2024 Accepted: 31 July 2024
Published online: 07 August 2024

References

1. M. Ulhaq, I.V. Bajić, Learned point cloud compression for classification, in *IEEE 25th International Workshop on Multimedia Signal Processing*. (2023)
2. A.F. Guarda, N.M. Rodrigues, M. Ruivo, L. Coelho, A. Seleem, F. Pereira, IT/IST/IPLEIRIA response to the call for proposals on JPEG pleno point cloud coding (2022), Preprint at arXiv arXiv:2208.02716
3. Y. Eldar, M. Lindenbaum, M. Porat, Y.Y. Zeevi, The farthest point strategy for progressive image sampling. IEEE Trans. Image Process. **6**(9), 1305–1315 (1997)
4. Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, Learning semantic segmentation of large-scale point clouds with random sampling. IEEE Trans. Pattern Anal. Mach. Intell. **44**(11), 8338–8354 (2021)
5. X. Wu, Y. Lao, L. Jiang, X. Liu, H. Zhao, Point transformer v2: grouped vector attention and partition-based pooling. Adv. Neural Inf. Process. Syst. **35**, 33330–33342 (2022)
6. I. Lang, A. Manor, S. Avidan, SampleNet: differentiable point cloud sampling, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2020), pp. 7578–7588
7. C. Wen, B. Yu, D. Tao, Learnable skeleton-aware 3D point cloud sampling, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2023), pp. 17671–17681(2023)
8. Y. Lin, Y. Huang, S. Zhou, M. Jiang, T. Wang, Y. Lei, DA-Net: density-adaptive downsampling network for point cloud classification via end-to-end learning, in *2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*. (IEEE, 2021), pp. 13–18
9. X. Wang, Y. Jin, Y. Cen, T. Wang, B. Tang, Y. Li, LighTN: light-weight transformer network for performance-overhead tradeoff in point cloud downsampling. IEEE Trans. Multimed. (2023). https://doi.org/10.1109/TMM.2023.3318073
10. C.R. Qi, H. Su, K. Mo, L.J. Guibas, PointNet: deep learning on point sets for 3D classification and segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2017), pp. 652–660
11. C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet++: deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inf. Process Syst. **30**, 5099–5108 (2017)
12. W. Wu, Z. Qi, L. Fuxin, PointConv: deep convolutional networks on 3D point clouds, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2019) pp. 9621–9630
13. H. Thomas, C.R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, L.J. Guibas, KPConv: flexible and deformable convolution for point clouds, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. (2019), pp. 6411–6420
14. W. Wu, L. Fuxin, Q. Shan, PointConvfFormer: revenge of the point-based convolution, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2023), pp. 21802–21813
15. H. Zhao, L. Jiang, J. Jia, P.H. Torr, V. Koltun, Point transformer, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. (2021), pp. 16259–16268
16. M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R.R. Martin, S.-M. Hu, PCT: point cloud transformer. Comput. Vis. Med. **7**, 187–199 (2021)
17. Z. Li, X. Tang, Z. Xu, X. Wang, H. Yu, M. Chen et al., Geodesic self-attention for 3D point clouds. Adv. Neural Inf. Process. Syst. **35**, 6190–6203 (2022)
18. H. Fan, Y. Yang, M. Kankanhalli, Point 4D transformer networks for spatio-temporal modeling in point cloud videos, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2021), pp. 14204–14213
19. Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph CNN for learning on point clouds. ACM Trans. Gr. **38**(5), 1–12 (2019)
20. A. Seleem, A.F. Guarda, N.M. Rodrigues, F. Pereira, Deep learning-based compressed domain point cloud classification, in *2023 IEEE International Conference on Image Processing (ICIP)*. (IEEE, 2023), pp. 2620–2624
21. T. Le, Y. Duan, Pointgrid: a deep network for 3D shape understanding, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2018), pp. 9204–9214
22. C. Fu, G. Li, R. Song, W. Gao, 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), pp. 625–633
23. W. MDGC, G-PCC codec description v9. MPEG-3DG. G-PCC Codec Description v9. ISO/IEC JTC1/SC29/WG7 N0011 (2020)
24. K. You, P. Gao, Q. Li, IPDAE: improved patch-based deep autoencoder for lossy point cloud geometry compression, in *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis*. (2022), pp. 1–10
25. B. Liu, S Li., X. Sheng, L. Li, D. Liu, Joint optimized point cloud compression for 3D object detection, in *2023 IEEE International Conference on Image Processing (ICIP)*. (IEEE, 2023), pp. 1185–1189
26. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł Kaiser, I. Polosukhin, Attention is all you need. Adv. Neural Inf. Process. Syst. **30**, 5998–6008 (2017)
27. Y. Pang, W. Wang, F.E. Tay, W. Liu, Y. Tian, L. Yuan, Masked autoencoders for point cloud self-supervised learning, in *European conference on computer vision*. ed. by S. Avidan, G. Brostow, M. Cissé, G.M. Farinella, T. Hassner (Springer, Berlin, 2022), pp.604–621
28. X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, J. Jia, Stratified transformer for 3d point cloud segmentation, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2022), pp. 8500–8509
29. J.G. March, L.S. Sproull, M. Tamuz, Learning from samples of one or fewer. Organ. Sci. **2**(1), 1–13 (1991)
30. X. Wang, Y. Jin, Y. Cen, C. Lang, Y. Li, PST-Net: point cloud sampling via point-based transformer, in *Image and Graphics: 11th International Conference, ICIG 2021, Haikou, China, August 6–8, 2021, Proceedings, Part III 11*. (Springer, 2021), pp. 57–69

31. C. Wu, J. Zheng, J. Pfrommer, J. Beyerer, Attention-based point cloud edge sampling, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2023), pp. 5333–5343
32. Y. He, X. Ren, D. Tang, Y. Zhang, X. Xue, Y. Fu, Density-preserving deep point cloud compression, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (2022), pp. 2333–2342
33. J. Canny, A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach Intell. **6**, 679–698 (1986)
34. J. Ballé, D. Minnen, S. Singh, S.J. Hwang, N. Johnston, Variational image compression with a scale hyperprior (2018), Preprint at arXiv arXiv:1802.01436
35. N. Tishby, F.C. Pereira, W. Bialek, The information bottleneck method, in *Proceedings of Annual Allerton Conference on Communication, Control and Computing*. (2000)
36. Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3Dshapenets: a deep representation for volumetric shapes, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (2015), pp. 1912–1920
37. S. Perry, H.P. Cong, L.A. Silva Cruz, J. Prazeres, M. Pereira, A. Pinheiro, E. Dumic, E. Alexiou, T. Ebrahimi, Quality evaluation of static point clouds encoded using mpeg codecs, in *2020 IEEE International Conference on Image Processing (ICIP)*. (IEEE, 2020) pp. 3428–3432
38. S. Perry, JEG pleno point cloud coding common test conditions v3.2. ISO/IEC JTC1/SC29/WG1 N 86044 (2020)
39. G. Bjontegaard, Calculation of average PSNR differences between RD-curves. ITU SG16 Doc. VCEG-M33 (2001)
40. G. Bjontegaard, Improvements of the BD-PSNR model. VCEG-AI11 (2008)

Publisher's Note