

Supplementary Materials: ROI-Guided Point Cloud Geometry Compression Towards Human and Machine Vision

Anonymous Authors

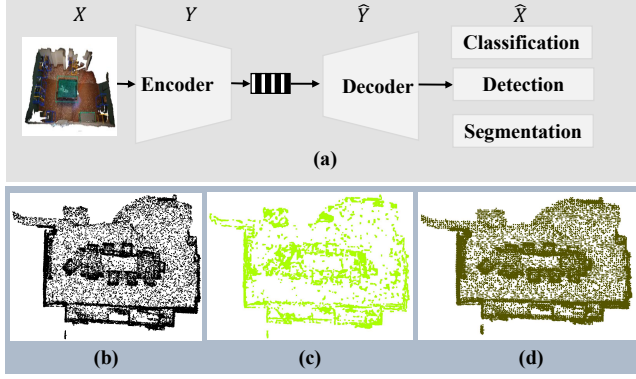


Figure 1: The standard process of compression for machine. (a) represents the point cloud compression pipeline for machine vision. (b) denotes the original point cloud of ScanNet, (c) presents the results of PCGCv2 coding, and (d) shows the outcomes of RPCGC coding.

The following materials will present our method’s motivation, modeling process, relevant computational pseudocode, experimental results, and visual analysis.

1 INTRODUCTION

Despite numerous end-to-end research efforts in point cloud compression, they primarily focus on fidelity optimization, as shown in Fig. 1 (a). However, in practical applications, the purpose of compression is to facilitate machine analysis. For instance, vehicles with LiDAR sensors in motion collect vast amounts of point clouds. Due to the limited space of storage devices, lossy compression is necessary during data collection and transmission in driving, leading to a significant loss of detailed information on the outdoor scenes. Similarly, in applications like indoor 3D reconstruction, the compression process can also result in the loss of semantic information in data. In video surveillance, data collected by cameras undergo compression before being transmitted to the cloud for analysis. These processes are unidirectional and irreversible, making it impossible to compensate for the loss of information in visual tasks through post-processing. Therefore, ensuring the performance of downstream tasks during compression becomes an urgent technical challenge. In end-to-end compression schemes, neural networks are employed in encoding modules, allowing the entire framework to be optimized jointly with different loss functions for specific tasks, which can simultaneously achieve machine perception and human vision optimization.

Firstly, we conduct object detection experiments on the ScanNet dataset after compressing its geometry information, using the RPCGC method alongside the advanced PCGCv2 [8] approach for data compression. We visualize the compressed data as illustrated in Fig. 1 (b), (c), and (d), representing the original point cloud, the

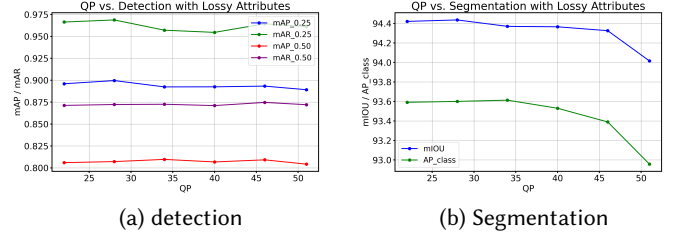


Figure 2: The performance analysis. (a) and (b) show the curves of geometry lossless and attribute lossy in downstream tasks influenced by different quantization values.

point cloud compressed by the PCGCv2, and the point cloud reconstructed after RPCGC compressor, respectively. The visualization results reveal a loss of geometry contours during compression. Then, we analyze the color information of the ScanNet to assess the impact on detection and segmentation tasks. We employ G-PCC lossless geometry compression combined with Region Adaptive Hierarchical Transform (RAHT) [3] for lossy attribute compression and utilize MinkUNet34C [2] as the segmentation network and FCAF3D [7] for point cloud detection. As demonstrated in Fig. 2 (a) and (b), our findings indicate that the impact of compressed point cloud color information on segmentation and detection tasks is negligible, even at high attribute compression rates. Therefore, this study primarily focuses on the compression of geometry information in point clouds.

2 RELATED WORKS

In this section, we explore approaches related to our research, concentrating on three primary areas: (1) The development of point cloud compression methods optimized for fidelity. (2) The advancement of image compression techniques specifically crafted for machine vision optimization. (3) The formulation of point cloud compression strategies to enhance machine vision capabilities. When dealing with compression tasks for various multimedia data, a diverse array of optimization strategies for machine vision is employed. For instance, as depicted in Fig. 3 (a), the approach involves compressing data and applying the compressed data in classification, detection, and segmentation. Fig. 3 (b) primarily focuses on extracting and analyzing semantic features from the compressed bitstream and using these insights for machine vision tasks. Fig. 3 (c) is specifically optimized for different tasks. Meanwhile, Fig. 3 (d) represents an approach that integrates the optimization of low-level compression tasks with high-level semantic analysis tasks, thereby creating a more holistic and efficient processing framework.

2.1 Image Compression for Machine

For example, Bai *et al.* [1] introduce a cloud-based, end-to-end image compression and classification model using modified Vision

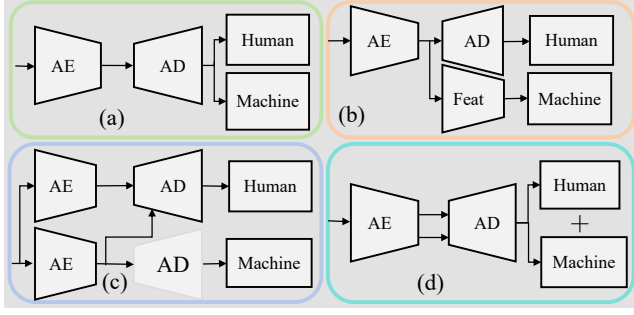


Figure 3: The compression pipelines for downstream tasks: (a) is serial mode. (b) is parallel mode. (c) represents multiple tasks with individual optimization. (d) denotes multiple tasks with joint optimization. “AE” and “AD” refer to the encoding and decoding processes, respectively, while “Feat” denotes the features of specific types of tasks. The term “Human” describes the reconstruction process, whereas “Machine” refers to downstream tasks.

Transformers [4] (ViT) that classifies images from compressed features, leveraging the Transformer’s ability to manage long-range information. The structure of [1] is similar to Fig. 3 (b). Liu *et al.* [5] propose a scalable image compression method for machine and human vision, featuring a pyramid representation for machine tasks and an optimized network for efficient encoding, balancing semantic accuracy, and signal reconstruction quality, the process of [5] is as shown in Fig. 3 (d).

2.2 Point Cloud Compression for Machine

The process is shown in Fig. 3 (a), Xie *et al.* [9] introduce a coding network utilizing sparse convolution, and design to extract semantic information for classification tasks concurrently. As shown in Fig. 3 (c), Liu *et al.* [6] introduce PCHM-Net, a new point cloud compression framework for human and machine vision, featuring a two-branch structure with a shared octree-based module and a point cloud selection module for sparse point optimization, coupled with a global feature aggregation-based classification module, achieving promising coding performance on various datasets.

3 METHODOLOGY

We represent the entire modeling process in the form of a Markov chain to facilitate a deeper understanding of the principles outlined in the RPCGC. Our encoding architecture is divided into two branches: the base layer stream and the enhancement layer stream. Simultaneously, we will preprocess the input point clouds using an RPN network. Subsequently, the generated mask will be weighted and applied to residual information and the loss function. The formulate process is illustrated in Fig. 4. Meanwhile, we have devised a masked residual weighting strategy. It utilizes the Feature Alignment Module (FAM) to expand the masks generated by the RPN into three-dimensional features, aligning them with the dimensions of the residual features. Subsequently, it applies the aligned result to the residual feature map. The pseudocode implementation of the entire weighting process is shown in Alg. 1, providing a clear insight into our weighting computation process.

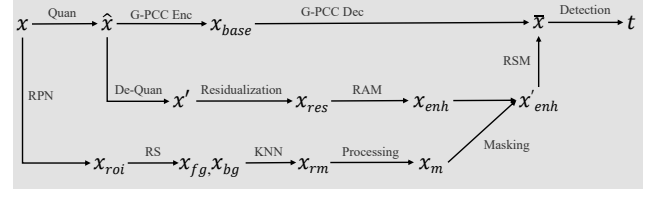


Figure 4: The pipeline chain of the RPCGC framework.

Algorithm 1 The mask generation and weighting in residual

```

1: min_bound ← inputs.min(axis=0)           ▷ xyz value
2: coords ← np.round(inputs - min_bound) * 50
3: coords, unq_idx ← np.unique(coords, return_index=True)
4: feats ← colors[unq_idx] - 0.5             ▷ rgb value
5: coords, feats ← ME.utils.sparse_collate([coords], [feats])
6: x ← ME.SparseTensor(feats, coords)
7: CLASS_IDS ← [1,2,3,...39]                ▷ category
8: bg ← [1, 2, 8, 9, 11, 16]                ▷ background
9: with torch.no_grad():
10:  soutput ← RPN(x)
11:  pred ← soutput.F.max(1).cpu().numpy()
12:  class ← np.array([CLASS_IDS[l] for l in pred])
13:  mask ← [1 if value in bg else 2 for value in class]
14:  x_coarse ← scale_sparse_tensor_batch(x, scaling)
15:  distance ← compute_nearest_neighbor(x.C, x_coarse.C)
16:  for j in range(len(distance)) do
17:    x_coarse_mask.append(mask[distance[j]])
18:  end for
19:  expend_mask ← torch.tensor(x_coarse_mask).unsqueeze(1)
20:  weight ← F.relu(Fc(expend_mask))
21:  weight ← Convs(weight.unsqueeze(0).transpose(1,2))
22:  feature ← Residual.F * weight + Residual.F

```

3.1 Ablation Study

Meanwhile, we categorize the post-compression detection results into four levels and conduct an in-depth analysis at different bitrates as shown in Fig. 5 (i) of the main paper: (1) $0 < \text{bpp} < 0.5$: Within this interval, the number of reconstructed point clouds is limited, retaining only the approximate location information of the point clouds, while detailed contour information suffers a significant loss. Therefore, detection performance seriously decreases under low bitrate conditions, characterized by a very low detection rate. (2) $0.5 < \text{bpp} < 1.0$: As the bitrate gradually increases, the quality of point cloud reconstruction improves, and the constraint information in the detection frame area becomes more abundant, thereby gradually enhancing detection performance. (3) $1.0 < \text{bpp} < 1.5$: In this range, with further increases in bitrate, the contours within the detection area of the point clouds essentially take shape. At this point, the Group-Free detection network begins to show insensitivity to the increase in the number of point clouds within the ROI, and the detection performance gradually approaches its limit. (4) $1.5 < \text{bpp} < 2.5$: The detection performance has peaked at this stage, and the detection results of point clouds within the ROI area no longer undergo notable changes.

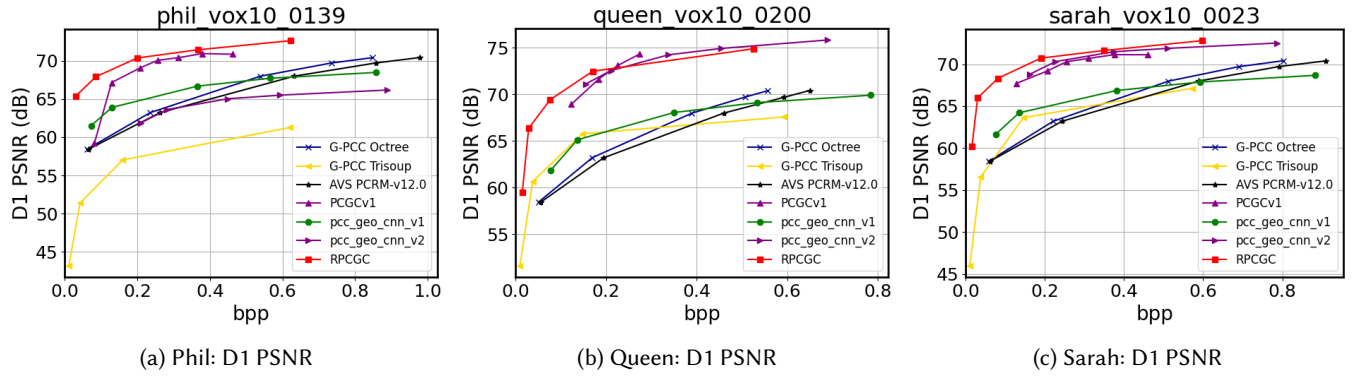


Figure 5: The RD curve of our proposed RPCGC and other representative methods in MPEG dataset.

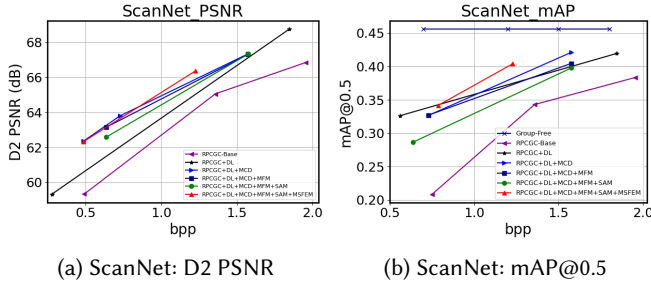


Figure 6: The ablation study for RPCGC in ScanNet dataset, “DL” represents Detection Loss, “MCD” refers to masking Chamfer Distance Loss, “MFM” signifies masking of the Residual Feature Map, “SAM” denotes the Semantic-Aware Attention Module, and “MSFEM” stands for the Multi-Scale Feature Extraction Module.

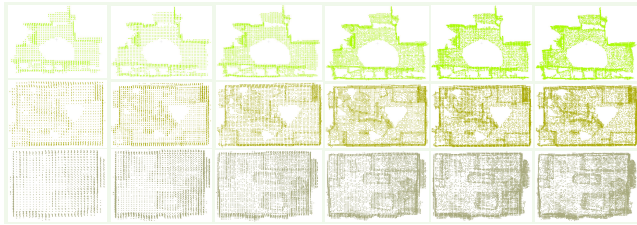


Figure 7: The reconstructed results of ScanNet via the RPCGC framework. The quantization ranges from 0.14 to 0.48. Each row corresponds to a distinct point cloud, showcasing the quality across six bitrates.

As shown in the Fig. 5, to evaluate the generalization capability of our approach, we plotted the RD curves based on the D2 PSNR indicator using datasets such as Phil, Queen, and Sarah. These datasets are commonly used standard test sets in MPEG. Meanwhile, the deep learning-based point cloud compression methods pcc_geo_cnn_v1, pcc_geo_cnn_v2, and PCGCv1 all adopt an autoregressive coding strategy. The graph shows that our method significantly outperforms existing coding schemes at low bitrates, while exhibiting marginal gains at high bitrates.

Fig. 6 (a) illustrates the D2 PSNR curves plotted using different strategies such as DL, MCD, MFM, SAM, and MSFEM, indicating our method’s advantageous compression performance at high bitrates. Fig. 6 (b) presents the detection performance at mAP@0.5 threshold. The curves demonstrate that the combined optimization of detection and compression tasks significantly improves detection performance at low bitrates. Meanwhile, the SAM structure guides the feature extraction process for detection tasks at higher bitrates, enhancing detection performance.

Furthermore, Fig. 7 illustrates the visualization outcomes of the RPCGC algorithm applied to ScanNet data across varying bitrates. This depiction aids in understanding the quality of point cloud reconstruction across different bitrates. The illustration reveals that lower bitrates lead to a significant loss of overall outline information in the point cloud, consequently lowering the detection rate of the model. Conversely, higher bitrates enable the complete reconstruction of object information. Consequently, learning-based methods tend to be most effective at higher bitrates.

REFERENCES

- [1] Yuanchao Bai, Xu Yang, Xianming Liu, Junjun Jiang, Yaowei Wang, Xiangyang Ji, and Wen Gao. 2022. Towards End-to-End Image Compression and Analysis with Transformers. In *AAAI Conference on Artificial Intelligence*, Vol. 36. 104–112.
- [2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 2019. 4D Spatio-Temporal Convnets: Minkowski Convolutional Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3075–3084.
- [3] Ricardo L De Queiroz and Philip A Chou. 2016. Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform. *IEEE Transactions on Image Processing* 25, 8 (2016), 3947–3956.
- [4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xi-aohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv preprint arXiv:2010.11929* (2020).
- [5] Kang Liu, Dong Liu, Li Li, Ning Yan, and Houqiang Li. 2021. Semantics-to-Signal Scalable Image Compression with Learned Reversible Representations. *International Journal of Computer Vision* 129, 9 (2021), 2605–2621.
- [6] Lei Liu, Zhihao Hu, and Jing Zhang. 2023. PCHM-Net: A New Point Cloud Compression Framework for Both Human Vision and Machine Vision. In *IEEE International Conference on Multimedia and Expo*. IEEE, 1997–2002.
- [7] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. 2022. Fcaf3d: Fully Convolutional Anchor-Free 3D Object Detection. In *European Conference on Computer Vision*. Springer, 477–493.
- [8] Jianqiang Wang, Dandan Ding, Zhu Li, and Zhan Ma. 2021. Multiscale Point Cloud Geometry Compression. In *Data Compression Conference*. IEEE, 73–82.
- [9] Liang Xie, Wei Gao, and Huiming Zheng. 2022. End-to-End Point Cloud Geometry Compression and Analysis with Sparse Tensor. In *International Workshop on Advances in Point Cloud Compression, Processing and Analysis*. 27–32.