A task-driven sampling method based on graph convolution for 3D point cloud compression

Yakun Yang, Anhong Wang*, Donghan Bu, Hao Jing,
School of Electronic Information and Engineering, TaiYuan University of Science and Technology, Taiyuan 030024, China
E-mail:yyk@tyust.edu.cn，wah_ty@163.com

The previous point cloud compression methods only consider reducing the amount of data. However, in applications such as autonomous driving, the compression methods not only require smooth transmission, but also improve the efficiency of downstream tasks. To this end, we propose a task-driven sampling network based on graph convolution to achieve point cloud compression and recovery. First, we present a task-driven downsampling network based on graph convolution to compress the point cloud. Then, we present an upsampling network based on graph convolution to enhance and recover the point cloud. In order to optimize the compressed point cloud for task, we add the task loss to loss function for end-to-end training. Experiments for point cloud classification task on ModelNet40 dataset show that the compressed point cloud obtained through our network can achieve higher classification accuracy compared to other similar methods, and the reconstructed point cloud can further improve classification accuracy.

Introduction: 3D point clouds can provide rich geometric and shape information. They are widely used in fields such as autonomous driving, virtual/augmented reality, and robotics. However, their large data volume, irregular structure, and sparsity make transmission and processing complex. Point cloud compression is necessary to save storage space, reduce the transmission bandwidth and communication load.

Currently, the Moving Picture Experts Group (MPEG) has proposed and developed the point cloud compression standard, named Geometry-based Point Cloud Compression (G-PCC) and Video-based Point Cloud Compression (V-PCC) [1]. The former processes static point clouds, representing unstructured point cloud data in an octree structure. The latter processes dynamic point clouds, mapping 3D point cloud into a 2D data format and then applying the 2D High Efficiency Video Coding(HEVC) to encode the projection plane. In addition, Google has developed the Draco [2] based on a k-d tree structure to compress point cloud. Since these methods mostly rely on hand-crafted coding strategies and cannot be implicitly optimized end-to-end, despite their excellent compression performance, there is still a large amount of redundant information.

With the great success of deep learning in point cloud, deep learning-based compression methods have the potential as a new compression standard. They can better adapt to the complex structure of point cloud, and obtain non-linear transforms at the encoder and decoder. As [3] used autoencoder to improve performance significantly compared with MPEG standard algorithms. According to the organization and representation of point cloud, deep learning-based compression methods are classified into point-based [4-5] and voxel-based methods [6-7]. Point-based methods have low computational complexity but poor reconstruction quality. They are affected by the sparsity of point cloud, and cannot handle large-scale LiDAR point cloud scenes. The voxel-based methods divide the point cloud into a voxel grid, ignoring the sparsity of the point cloud and allowing the use of a variety of geometric and spatial information, but they also leading to high computational and memory costs that grow cubically as the resolution increases.

The above traditional and deep learning-based point cloud compression methods only consider compression metrics such as bitrate and distortion. They do not combine downstream tasks, thus reducing the practical application efficiency of the methods.

To address these issues, this paper proposes a task-driven sampling method for point cloud, using a downsampling network for compression and an upsampling network for enhancement and recovery. Due to the graph structure is more suitable for processing unstructured non-Euclidean data, so our network will combine graph convolution to represent and learn 3D point cloud.

Method: We proposed a task-driven sampling method for point cloud compression and reconstruction, as shown in Fig.1.

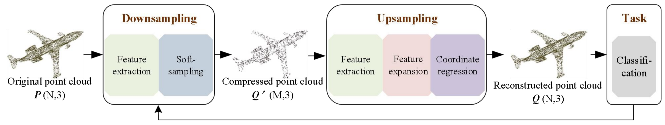

Fig. 1 *The structure of task-driven sampling method for 3D point cloud compression*

This model consisted of three networks: downsampling, upsampling, and application tasks. Given a 3D point cloud P containing N points, the task-driven downsampling network achieved simplification and compression of point cloud. The compressed point cloud Q' can be optimized for the application task by introducing a task loss in the joint loss function. After storage and transmission, the upsampling network enhanced and recovered the point cloud. Finally, the reconstructed point cloud Q was applied to various point cloud tasks. Where r is the sampling rate and $M = N/r$ is the number of compressed point cloud, $M < N$.

As we know, Graph is a data structure for modeling objects (nodes) and their relationships (edges). Its powerful representational capabilities have led to widely uesd in social networks, biomolecules and other fields. In addition, its unique non-Euclidean data structure is well suited for sparse, irregular 3D point cloud. Convolutional Neural Network (CNN) can effectively extract spatial features, but it can only be used in regular data structures. The Graph Convolution Network (GCN) [8], which combines the graph and CNN, is now widely used in various point cloud learning networks. In this paper, we use graph convolution to construct the downsampling and upsampling networks.

As shown in Fig.2, the task-driven downsampling network based on graph convolution consisted of feature extraction unit and soft sampling unit.

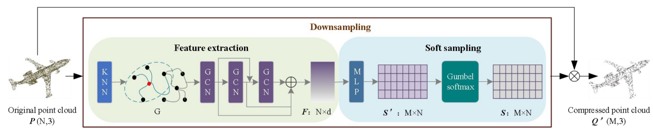

Fig. 2 *The structure of task-drived downsampling network based on graph convolution network*

In Feature extraction unit, we used a densely connected GCN module to learn the features of point cloud. As shown in the green part of Fig.2, a local graph structure G was constructed for each point according to the K Nearest Neighbor (KNN) algorithm firstly. Secondly, the point cloud features were extracted by multiple stacked GCNs. Finally, the features learned by the densely connected GCNs were summed to obtain the d-dimensional multi-scale point cloud features F.

Next, the sampling process for point cloud can be expressed by Eq.1:

$$Q' = S \times P \tag{1}$$

Where P is the original point cloud, S is the sampling matrix, and Q' is the compressed point cloud obtained by downsampling. The ideal sampling matrix S contains only 0 and 1, however this discrete non-differentiable matrix cannot be trained end-to-end in a deep neural network. We proposed to use a soft sampling matrix to approximate the ideal sampling matrix, where each element in the soft sampling matrix is not 0 or 1, but a number between 0 and 1, and the closer it is to 0 or 1, the better. As shown in the blue section of Fig.2, the soft sampling matrix was learned using the MLP and the Gumbel-Softmax module. First, given the number of compressed point clouds M, the features F obtained by the feature extraction unit are passed through the MLP to get a correlation matrix S'. In order to make the elements in (0,1), we need to normalize it. As shown in Eq.2, it was implemented using the Gumbel-Softmax[9].

$$\sigma(z_i) = \frac{e^{z_i/t}}{\sum_{j=1}^{K} e^{z_j/t}} \tag{2}$$

Gumbel-Softmax is a Softmax function with parameter control. The higher the t, the 0-1 distribution is smoother, and the lower the t, the distribution is closer to a discrete one-hot. The soft sampling matrix can be approximated to the ideal sampling matrix in the training process by gradually decreasing the parameter t. Finally, the original point cloud P was multiplied by the learned soft sampling matrix S to obtain the downsampled compressed point cloud.

In the task-driven downsampling network, we designed a joint loss function consisting of the task loss, the sampling loss, and the soft sampling constraint loss, as shown in Eq.3.

$$L_{down} = \lambda_{task} L_{task} + \lambda_{samping} L_{sampling} + \lambda_{constraint} L_{constraint} \tag{3}$$

where λ is the parameter for balancing each item.

The compressed point cloud was stored, transmitted, and then enhanced for recovery by an upsampling network. As shown in Fig.3, this network contained feature extraction unit, feature expansion unit, and coordinate regression unit.

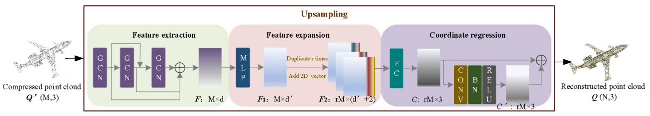

Fig. 3 *The structure of upsampling network based on graph convolution network*

In Feature extraction unit, we still used the graph convolution network. The densely connected graph convolutional layer can learn the point cloud's local and global geometric information from different levels of detail.

Since points and features are interchangeable, we can regard M points with rd-dimensional features as rM points with d-dimensional features. As shown in the orange part of Fig.3, the Feature expansion unit can increase the number of points. We used MLP to get features F_1 of M points and then copy r times to obtain the d_1-dimensional features of rM points. However, such simple convolution and replication will make the generated points too similar to get a uniform and dense point cloud. Therefore, we used the 2D grid mechanism in the FoldingNet [10] to generate a unique 2D vector for each feature, which was then added to the features of each point to obtain different features. It increased the diversity of the point cloud features and thus made the individual points subtly different.

The coordinate regression unit regressed the d+2 dimensional feature F_2 of rM points into 3D coordinates and obtained the reconstructed point cloud Q. Since absolute coordinates vary more than relative offsets in 3D space, and the residuals can highlight slight variations, we proposed using a residual correction module to obtain accurate point cloud coordinates. As shown in the purple part of Fig.3, F_2 was transformed into the 3D coordinates of rM points through a fully connected layer. To reduce the effect of noise and generate a dense point cloud, we added a residual correction unit consisting of a Convolution layer, a Batch Normalization layer, and a Rectified Linear Unit, which regresseed the residual offset of each point's position. Finally, add the offsets obtained by refinement to the coarse coordinates to get the precise 3D coordinates.

The loss function of the upsampling network consisted of the sampling loss and the regularization loss. The sampling loss included the reconstruction loss and the repulsion loss, as shown in Eq.4.

$$L_{up} = \lambda_{rec} L_{rec} + \lambda_{rep} L_{rep} + \lambda_{reg} L_{reg} \tag{4}$$

where λ is the parameter for balancing each item.

Experimental results: We used a computer equipped with a RTX8000 GPU to conduct experiments. For simple implementation, we chose the point cloud classification as downstream task and PointNet [11] as the classification task network. The task evaluation metric is classification accuracy. We used the modelNet40 [12] as the dataset for point cloud classification, which contains 12311 3D objects in 40 categories, of which 9843 were used for training and 2468 for testing. Each object was first uniformly sampled to 1024 points before training. When training the task-driven downsampling network, the parameter settings were kept consistent with S-Net. When training the upsampling network, the parameter settings were kept the same as those of PU-GCN.

First, we compared three task-oriented methods, S-Net [13], SampleNet [14], MOPS-Net [15] with our network proposed in this paper. r is the downsampling rate. When r=1, the original point cloud with 1024 points, and when r=2, 4, 8, 16, 32, 64, 128, the corresponding number of the compressed point clouds are 512, 256, 128, 64, 16 and 8 respectively. The classification accuracy of the different methods at each sampling rate is shown in Fig.4.

As seen from Fig.4, our downsampling network can achieve higher classification accuracy at all sampling rates, especially when the sampling rate is large, the advantage is more prominent. In addition, when the number of point cloud is compressed to 1/32 of the original size, our downsampling network can still achieve more than 80% classification accuracy, which can get better classification results and satisfy most application scenarios. This is very friendly to lightweight tasks and facilitates the transmission and storage of the point cloud.

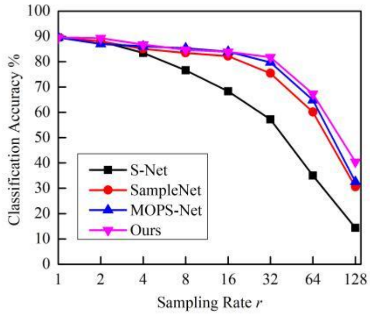

Fig. 4 *The classification accuracy of different methods at each sampling rate r*

To further improve the performance of the task, we used an upsampling network to enhance and recover the compressed point cloud. we used the graph convolution-based upsampling network to obtain a reconstructed point cloud for the classification task. Fig.5 shows the classification accuracy of the compressed point cloud after random downsampling, and the reconstructed point cloud using different upsampling methods such as PU-Net [16]、PU-GAN [17]、PU-GCN [18] to recover the compressed point cloud. The black dashed line shows the classification accuracy of the compressed point cloud after random sampling, while the rest of the colored lines show the classification accuracy of the reconstructed point cloud after the different upsampling methods.

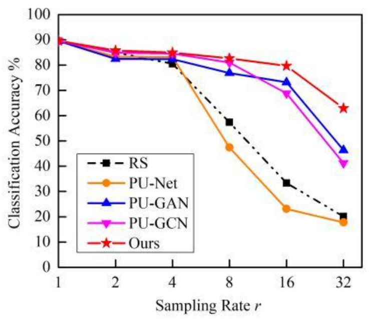

Fig. 5 *The classification accuracy of random downsampled point cloud and different upsampling reconstructed point cloud at each sampling rate*

It can be seen that the classification accuracy of the compressed point cloud after random sampling gradually decreases as the sampling rate becomes larger, and after the enhancement and recovery by upsampling, the classification accuracy is improved.The larger the sampling rate, the more pronounced the improvement, and our upsampling network is the most improved. What draws attention is that the classification accuracy of the reconstructed point cloud enhanced with PU-Net is even lower than that of the compressed point cloud when the sampling rate is larger. This is because the fact that the recovery of the point cloud from fewer points amplifies the detail defects, which affects the overall point cloud quality and application task’s performance.

The above experiment results show that the upsampling network proposed in this paper can effectively improve the quality of compressed point cloud, and at the same time, can further improve the performance of subsequent application tasks.

Previous results have demonstrated the effectiveness of our task-driven downsampling network and our upsampling network, both of which are combined to form a task-driven compression and reconstruction model for point cloud. Compared with the non-task-driven compression and reconstruction, the results were shown in Table.1. The second column was the classification accuracy of the compressed point cloud after traditional random downsampling, and the third column was the classification accuracy of the reconstructed point cloud after upsampling. The fourth column shown the classification accuracy of the compressed point cloud after task-driven downsampling, and the fifth column shown the classification accuracy of the reconstructed point cloud after upsampling to recover.

Table 1: The classification accuracy of compressed and reconstructed point cloud with non-task-driven and task-driven model at each sampling rate

Sampling rate	RS	Upsampling	Task-driven downsampling	Upsampling
2	85.06	85.76	89.31	89.44
4	80.54	84.92	85.80	96.76
8	57.38	82.65	84.27	86.18
16	33.35	79.68	83.93	85.55
32	20.13	62.93	81.76	83.24

We can see that the task-driven downsampling network proposed in this paper dramatically improves the classification accuracy compared to the traditional downsampling compression method with the same sampling rate. In addition, the reconstructed point cloud's classification accuracy after upsampling is further improved, and the improvement is more pronounced when the sampling rate is larger.

These experimental results show that the task-driven downsampling network proposed in this paper can achieve compression of the point cloud, which not only reduces the amount of point cloud but also guarantees the performance of task. At the same time, the upsampling network proposed in this paper can enhance and recover of compressed point cloud, further improving point cloud quality and task performance. The whole task-driven point cloud compression and reconstruction model is practical and feasible.

Conclusion: This paper proposes a task-driven sampling network based on graph convolution for the compression and reconstruction of 3D point cloud. The task-driven downsampling network containing a graph feature extraction unit and a soft sampling unit to generate the compressed point cloud. The graph-based upsampling network with a residual correction unit enhances and recovers these compressed point cloud. We construct a joint loss function with the task loss for end-to-end training to ensure that the sampled point cloud can be optimized for the downstream task. Experiments on the ModelNet40 dataset show that the proposed method not only reduces data volume but also ensures high classification accuracy, without affecting the performance of subsequent applications. Moreover, the reconstructed point cloud after enhancement and recovery can further improve the performance of the task. In the future, we will combine more efficient methods for point cloud learning, such as Transformer, while the proposed network will be applied to other point cloud tasks, such as segmentation and object detection.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (No.62072325), The University Science and Technology Innovation Project of Shanxi(No.2022L326).

References

1. Schwarz, Sebastian., et al.: 'Emerging MPEG standards for point cloud compression'. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2018, 9(1), pp.133-148.
2. 'Google Draco 3D graphics compression[EB/OL]'. https://google.github.io/draco/. (2018)
3. Quach, Maurice., Giuseppe, Valenzise., and Frederic, Dufaux.: 'Learning convolutional transforms for lossy point cloud geometry compression'. Proceedings of IEEE international conference on image processing. Piscataway:IEEE Press, 2019, pp.4320-4324.
4. Huang, Tianxin., and Liu, Yong.: '3d point cloud geometry compression on deep learning'. Proceedings of the 27th ACM international conference on multimedia. Piscataway:IEEE Press, 2019, pp.890-898.
5. Wiesmann, Louis., et al.: 'Deep compression for dense point cloud maps'. *IEEE Robotics and Automation Letters*, 2021, 6(2), pp.2060-2067.
6. Wang, Jianqiang., et al.: 'Lossy point cloud geometry compression via end-to-end learning'. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, 31(12), pp.4909-4923.
7. Que, Zizheng., Guo, Lu., and Dong, Xu.: 'Voxelcontext-net: An octree based framework for point cloud compression'. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway:IEEE Press, 2021, pp.6042-6051.
8. Lin, Zhihao., Huang, Shengyu., and Yu-Chiang Frank Wang.: 'Learning of 3d graph convolution networks for point cloud analysis'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, 44(8), pp.4212-4224.
9. Yang, Jiancheng., et al.: 'Modeling point clouds with self-attention and gumbel subset sampling'. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. Piscataway:IEEE Press, 2019, pp.3323-3332.
10. Yang, Yaoqing., et al.: 'Foldingnet: Point cloud auto-encoder via deep grid deformation'. Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway:IEEE Press, 2018, pp.206-215.
11. Qi, Charles R., et al.: 'Pointnet: Deep learning on point sets for 3d classification and segmentation'. Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway:IEEE Press, 2017, pp.652-660.
12. Wu, Zhirong., et al.: '3d shapenets: A deep representation for volumetric shapes'. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway:IEEE Press, 2015, pp.1912-1920.
13. Dovrat, Oren., Itai, Lang., and Shai, Avidan.: 'Learning to sample'. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway:IEEE Press, 2019, pp.2760-2769.
14. Itai, Lang., Asaf, Manor., and Shai, Avidan.: 'Samplenet: Differentiable point cloud sampling'. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway:IEEE Press, 2020, pp.7578-7588.
15. QIAN Y, HOU J, ZHANG Q, et al. MOPS-Net: A matrix optimization-driven network fortask-oriented 3D point cloud downsampling[J]. arXiv preprint arXiv:2005.00383, 2020.
16. Yu, Lequan., et al.: 'Pu-net: Point cloud upsampling network'. Proceedings of the IEEE conference on computer vision and pattern recognition. Piscataway:IEEE Press, 2018, pp.2790-2799.
17. Li, Ruihui., et al.: 'Pu-gan: a point cloud upsampling adversarial network'. Proceedings of the IEEE/CVF International Conference on Computer Vision. Piscataway:IEEE Press, 2019, pp.7203-7212.
18. Qian, Guocheng., et al.: 'Pu-gcn: Point cloud upsampling using graph convolutional networks'. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway:IEEE Press, 2021, pp.11683-11692.