SPC-NET: A NEW SCALABLE POINT CLOUD COMPRESSION FRAMEWORK FOR BOTH MACHINE AND HUMAN VISION TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, point cloud process and analysis have attracted increasing attention in various machine vision tasks. Therefore, some point cloud compression algorithms are developed. However, such compression algorithms are developed for human vision while most of the point cloud data will be used for automated point cloud analysis (*e.g.,* detection of abnormal event and early warning in autonomous driving) and may not be seen by humans. To this end, we design a new scalable point cloud compression framework (SPC-Net) for both machine and human vision tasks, in which a scalable bit-stream will be used to describe the point cloud for both machine vision and human vision tasks. For machine vision tasks, only part of the bit-stream will be transmitted for bit-rate saving, while the full bit-stream will be transmitted when used for the human vision task. Additionally, we propose a new octree depth level predictor to automatically predict the optimal depth level in order to control the bit-rate cost for the machine vision tasks. As a result, for simple objects/scenarios, we will use fewer depth levels with less bits for the machine tasks, while for complex objects/scenarios, we prefer deeper depth levels of octree with more bits for machine tasks comprehensive. Experimental results on different datasets (*e.g.,* ModelNet10, ModelNet40, ShapeNet and ScanNet) demonstrate that our proposed scalable point could compression framework SPC-Net achieves better performance on the machine vision tasks (*e.g.,* classification, segmentation and detection) without degrading the performance of the human vision task.

1 INTRODUCTION

With the development of advanced 3D technologies, it has become easier to collect point clouds by using various types of 3D scanners including LiDARs and RGB-D cameras. Therefore, a huge amount of point cloud data has been collected and various point cloud related machine vision tasks like classification, segmentation and detection have attracted increasing attention. However, most point cloud analysis tasks take raw point cloud data as the input, which requires large bandwidth/storage for transmitting/storing huge massive point cloud data.

Recently, some point cloud compression frameworks Huang et al. (2020); Que et al. (2021) were proposed to save the bandwidth/storage for point cloud transmitting/storing. However, those point cloud compression frameworks are designed for human vision, which will thus degrade the performance in the machine vision tasks. Currently, the existing point cloud compression framework are not designed for the machine vision tasks. Some recent works Yang et al. (2021); Le et al. (2021); Song et al. (2021); Torfason et al. (2018) have explored the image coding for machine task by optimizing the network with additional loss functions for the machine vision tasks. However, the state-of-the-art point cloud compression algorithms like VoxelContext-NetQue et al. (2021) need to construct the octree and then compress it, in which the octree construction procedure is indifferentiable and thus we cannot directly add the loss function to improve the coding performance for the machine vision tasks. Therefore, it is necessary to design a new point cloud compression framework for both human and machine vision tasks.

In this work, we propose the first point cloud compression framework for both human and machine vision, Our framework follows the scalable coding paradigm, in which the full bit-stream will used for the human vision task, while only part of the bit-stream will be used for the machine vision tasks. For the human vision task, we take the start-of-the-art method VoxelContext-Net Que et al. (2021) as an example to compress the point cloud, in which the octrees are constructed and then compressed into bit-streams. For the machine vision tasks, we only transmit part of the bit-stream to reconstruct the first few depth-level of the octrees for bit-rate saving. Additionally, we propose the octree depth level predictor to predict the optimal depth-level of the octree for different scenarios when for coding for machine vision tasks. As a result, for simple objects/scenarios, we will use less depth level with less bits for bit-rate saving, while for complex objects/scenarios, we prefer deeper depth level of octree for more accurate prediction. Experimental results demonstrate that our proposed framework SPC-Net achieves promising results on various machine vision tasks without sacrificing the coding performance for the human vision task.

- In this work, We propose a new scalable point cloud compression framework for both machine vision and human vision tasks. To the best of our knowledge, this is the first point cloud compression method for both machine and human vision.
- We propose a new octree depth level predictor to predict the optimal depth of the octree used for the machine vision tasks, where deeper octree will be used for complex objects/scenarios for achieving more accurate prediction while shallow octree will be used for simple objects/scenarios for achieving less bit-rate cost.
- Comprehensive experimental results demonstrate that our proposed scalable point cloud coding framework achieves promising results without sacrificing the coding performance of the human vision task.

2 Related work

2.1 Point Cloud Compression for Human Vision

In the past few years, hand-crafted and learning-based point cloud compression methods Group (2021); Wang et al. (2021b); Biswas et al. (2020); Huang et al. (2020); Zhu et al. (2020); Que et al. (2021) have been proposed by transforming the point cloud data into tree representations for better compression.

Specifically, a few hand-crafted point cloud compression methods Group (2021); Schwarz et al. (2018); Google (2022) have been proposed. For example, the standard point cloud compression method G-PCC (geometry based point cloud compression) Group (2021) proposed by the MPEG group, which transforms point cloud data into the octree-structure before performing static point cloud compression.

In recent years, some learning-based point cloud compression methods Huang & Liu (2019); Zhu et al. (2020); Huang et al. (2020); Biswas et al. (2020); Que et al. (2021); Wang et al. (2021b;a) have achieved the state-of-the-arts performance. Huang *et al.* Huang et al. (2020) and Wang *et al.* Wang et al. (2021b) followed the learned image compression framework Ballé et al. (2017) to compress the voxelized point clouds. To reduce the bitrate, Biswas *et al.* Biswas et al. (2020) exploited the spatio-temporal relationships across multiple LiDAR sweeps by using a novel conditional entropy model. Based on Wang et al. (2021b), Wang *et al.* Wang et al. (2021a) used the lossless compressed octree and the lossy compressed point feature to further improve the coding performance. Que *et al.* Que et al. (2021) extended the framework by further exploiting the context information among neighbouring nodes and refining the 3D coordinate at the decoder side. Considering that VoxelContext-Net Que et al. (2021) is the state-of-the-art point cloud compression method, we use it as our baseline method for the human vision task.

All the existing methods compress the point cloud data for human perception, which is evaluated by the metrics like point-to-point PSNR and point-to-plane PSNR. However, unlike the 2-D images or videos, most point clouds are not purely collected for human perception. Instead, they are widely used for various real-world machine vision tasks, such as classification, segmentation, and detection, which is unfortunately not considered in there works

2.2 IMAGE COMPRESSION FOR BOTH MACHINE AND HUMAN VISION TASKS

To the best of our knowledge, there is no existing point cloud compression method for both machine and human vision. In this section, we first discuss the scalable image compression methods for both machine and human vision tasks, and then the other compression methods.

Scalable Methods. Both Choi *et al.* Choi & Bajić (2022)and Chen *et al.* Chen et al. (2021) performed scalable image compression by dividing the image bit-streams into different parts and transmitting one or more parts of the bit-streams for both machine or human vision tasks. Liu *et al.* Liu et al. (2021) proposed a scalable image compression method for define grained classification at different levels.

Other Methods. Yang *et al.* Yang et al. (2021) designed the image encoder by using the edge extraction algorithm, and the reconstructed images from the decoder achieve promising performance for both human vision and machine vision tasks. Le *et al.* Le et al. (2021) directly added the additional machine vision loss to the compression loss functions to improve the reconstructed image quality for the machine vision tasks. Song *et al.* Song et al. (2021) compressed the source image through a corresponding quality map produced from different machine vision tasks. Torfason *et al.* Torfason et al. (2018) combined the image compression network with the detection network, and directly extracted the detection related information from bit-stream without using an image decoder.

In summary, the above methods for machine vision methods are all lossy compression methods. The encoder extracts the helpful features from the images, the decoder reconstructs the images based on the encoded features, and the entropy model calculates the number of bits used for the features. Most methods can adjust the various parameters in the encoder and decoder based on the performance in machine vision tasks. Therefore, it is easy for the encoder to learn the representative image features for machine vision. However, most learning-based point cloud compression methods use a lossless compression network. Their encoders and decoders cannot be optimized to extract the useful features in the point cloud for machine vision, which hinders the development of the point cloud compression methods for the machine vision tasks.

In contrast to these works Liu et al. (2021); Chen et al. (2021); Choi & Bajić (2022); Yang et al. (2021); Le et al. (2021); Song et al. (2021); Torfason et al. (2018), we propose some new modules before and after the compression model to improve the machine vision performance while maintaining the fidelity for human vision by keeping the lossless point cloud compression model unchanged.

3 METHODOLOGY

3.1 THE FRAMEWORK

The overall structure of our scalable point cloud compression framework (SPC-Net) is shown in Figure 1(b). In this section, we will first introduce our method coding strategy. And then, each module in our framework will be introduced.

Scalable Coding Strategy. The point cloud data is commonly used for various machine vision tasks. Therefore, our SPC-Net is always used for machine vision tasks (*e.g.,* abnormal event detections to detect collision between the pedestrians and the vehicles) and the point cloud information is transformed along the solid arrows as shown in Figure 1 (b). If the human vision task must also be involved (*e.g.*, when the prediction results from the machine vision tasks like event detection are abnormal), our framework can provide a high quality reconstructed point cloud for humans further analysis. It should be mentioned that like the scalable coding method, to reconstruct the point clouds for the human vision task, we can reuse the bit-stream generated for the machine vision task, which can avoid duplicate bit transmission.

Octree Construction, Encoder, Decoder and Point Cloud Reconstruction. The octree construction module constructs the point cloud to octree. Octree is a tree-like data structure used to describe three-dimensional space. Each node of the octree represents a volume element of a cube, and each non-leaf node has eight child nodes. The volume of the parent node can be obtained by adding the volume elements represented by the eight child nodes together. And the black node in Figure 1 (a) means there are points in this cube, and the white node means empty cube without having any 3D point. Each octree is encoded as the bit-stream by using the 3D encoder. The decoder reconstructs

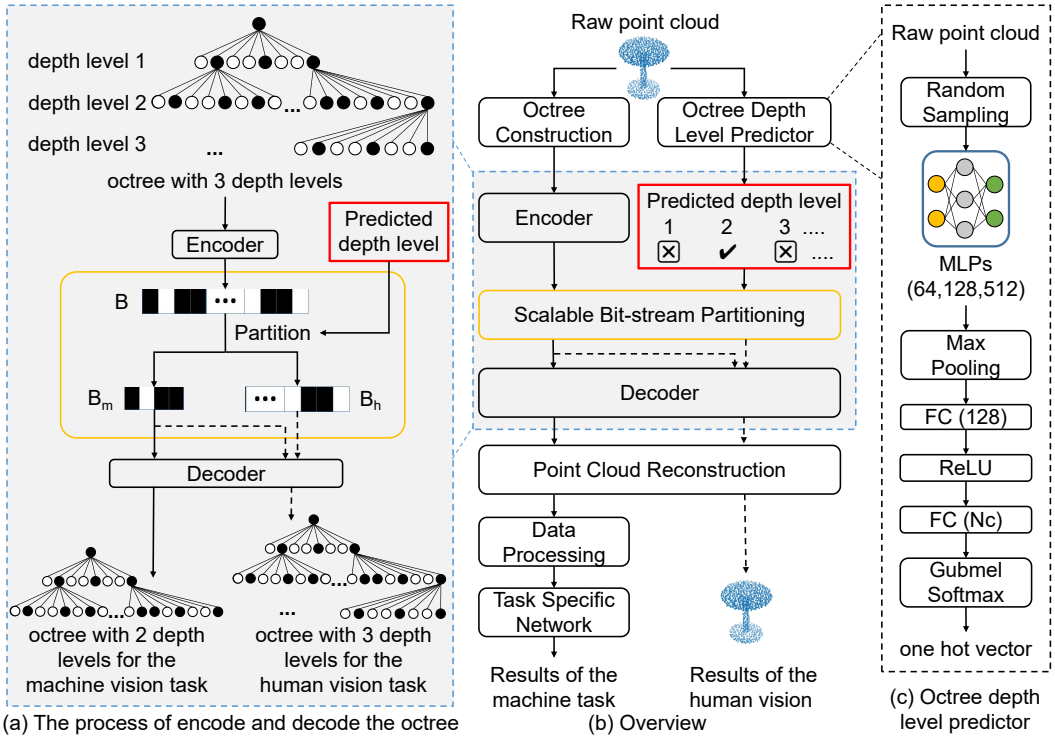

Figure 1: (a) The encoding and decoding process of the octree. $\mathbf{B}$, $\mathbf{B_m}$ and $\mathbf{B_h}$ denote the full bit-stream, the bit-stream for the machine vision task and the bit-stream for the rest depth level of the octree, respectively. (b) The overall architecture of our proposed scalable point cloud compression framework SPC-Net, which is designed for both machine vision and human vision. (c) Details of our proposed octree depth level predictor.

the bit-stream to octree. The point cloud reconstruction module then restore the point cloud from the octree. In this work, we task VoxelContext-Net Que et al. (2021) as an example and use the same design for all those modules, the details can be found in Appendix A.1.

Scalable Bit-stream Partitioning. Our scalable bit-stream partitioning module can split the full bit-stream to two parts bit-stream for different tasks. The details is shown in section 3.2.

Octree Depth level Predictor. Our octree depth level predictor is used to adaptively choose the octree depth for the machine vision tasks and can guide the full bit-steam splitting. The details of this module will be described in section 3.3.

Data Processing. The role of this module is to process point cloud data to compensate for the data difference between the output of the compressed network and the input of the machine task network. The details about this module are shown in Appendix A.1.

Task Specific Network. To adapt to a variety of situations in the point cloud based machine vision tasks, this module will use different networks for different machine vision tasks. For the classification task and the segmentation task, PointNet++ Qi et al. (2017) will be used in this module. For the detection task, VoteNet Qi et al. (2019) is adopted.

3.2 Scalable Bit-stream Partitioning

Although the reconstructed point cloud often achieves promising performance for the human vision task when using full bit-stream, it has plenty of redundant information for the machine vision tasks and thus it is less effective in terms of the bit-rate cost. Therefore, we design this scalable bit-stream partitioning method to split the bit-stream for both human and machine vision tasks.

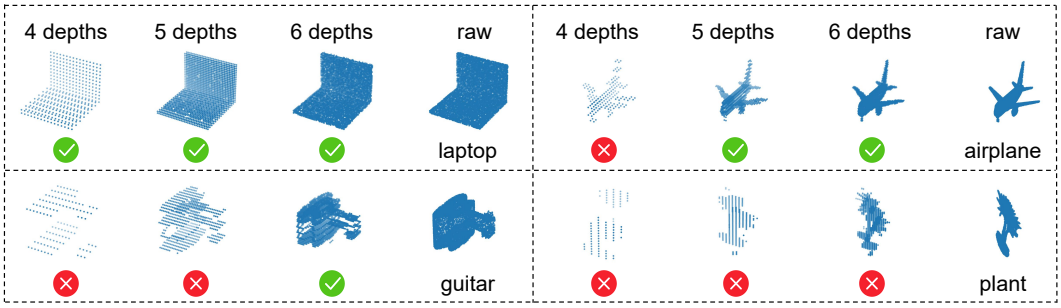

Figure 2: The classification results of a pre-trained PointNet++ model for point clouds reconstructed from different octree depth levels. "4 depths", "5 depths" and "6 depths" mean that the point cloud is reconstructed from the octrees with 4, 5 and 6 depth levels. "raw" means the raw point cloud. The truely or falsely predicted results of the classification task are shown under the point clouds.

Before introducing how to divide the bit-stream, we first introduce how to generate the point cloud bit-stream. Figure 1(a) shows the encoding and decoding process of the octree. During the encoding process, each octree is encoded from the lower depth level to the higher depth level. Therefore, the final full bit-stream can be expressed as $\mathbf{B} = (b_1, b_2, ..., b_n)$, where n is the maximum octree depth level and b_i represents the bit-stream from the ith depth level. At the decoder side, each octree will be reconstructed from the lower depth level to the higher depth level. The $(i+1)$th depth level of the octree can be reconstructed with the previously reconstructed octree which has i depth levels and the extra bits b_{i+1}. For example, with $b_1 \cup b_2$, we can reconstruct the octree with the first two depth levels, and with $b_1 \cup b_2 \cup b_3$ we can reconstruct the octree with the first three depth levels.

Based on the above octree encoding and decoding process, we can split the full bit-stream $\mathbf{B} = (b_1, b_2, ..., b_n)$ into two parts $\mathbf{B_m}$ and $\mathbf{B_h}$ according to the octree depth level. $\mathbf{B_m} = (b_1, b_2, ..., b_i)$ can be used to reconstruct the octree with the first i depth levels, which will be used for the machine vision tasks. $\mathbf{B_h} = (b_{i+1}, b_{i+2}, ..., b_n)$ can reconstruct the rest depth levels of the octree based on the reconstruction of the first i depth levels, which will be used for the human vision task. And the optimal splitting level index i is determined by the octree depth predictor for scalable bit-stream partitioning.

3.3 Octree Depth Level Predictor

The design of our octree depth level predictor is inspired by the well-trained machine vision tasks (*i.e.,* classification, segmentation, detection). We can often achieve reasonable results when using the reconstructed point cloud from the lower depth level octree as the input. Taking the classification results in Figure 2 as an example, some objects with simple shapes like laptop can be easily recognized when using the reconstructed point cloud reconstructed from the octree with 4 depth levels as the input, while other objects with complex shapes like guitar can only be recognized when using the reconstructed point cloud from the octree with 6 depth levels. Therefore, we can use the octree with lower depth levels to reduce bit-stream cost and thus we can save the storage space and the bandwidth.

To achieve this goal, we propose the octree depth predictor to decide the optimal depth level of the octree for the machine vision tasks, which can not only achieve the reasonable performance for the machine vision tasks but also reduce the bit-rate cost. In addition, the encoder side (*e.g.,* the RGB-D cameras or the LiDAR sensors) always do not have enough computing power and can not support the complex networks. Therefore, the networks (*e.g.,* PointNet++ and VoteNet) for handling the complex machine vision tasks are placed behind the decoder and not in the encoder side. As shown in Figure 1 (c), our octree depth level predictor is designed by using 3 layers MLP, and 2 fully connected layers, which is a simple network. To future reduce the computational complexity, we random sample 1024 points from the raw point cloud as the input of our octree depth level predictor for classification and segmentation tasks.

Our octree depth level predictor can select the optimal octree depth level for machine vision tasks from the raw point cloud global feature. According to the different characters in machine vision tasks (*e.g.,* the difficulty of classification), our octree depth predictor can generate n probabilities $\mathbf{p} = \{p_1, p_2, ..., p_n\}$ for n octree depth levels, and then choose the octree depth level with the highest probabilities.

However, the process of choosing the depth level of octree with the highest probability is non-differentiable, which makes the octree depth predictor unable to train. Therefore, we adopt the Gumbel Softmax Strategy Jang et al. (2017) to address this issue. First, we generate confidence score set $\hat{\mathbf{p}}$ from the probability set $\mathbf{p}$ with Gumbel noise as follows:

$$\hat{p}_i = p_i + G_i, i \in \{0, 1, ..., n\} \tag{1}$$

where $G_i = -\log(-\log \epsilon)$ is the standard gumbel noise, and ϵ is randomly sampled from a uniform distribution between 0 and 1. Therefore, we can generate the one-hot vector $\hat{\mathbf{h}} = [\hat{h}_0, \hat{h}_1, ..., \hat{h}_n]$, where $\hat{h}_i = 1$ if $i = \arg\max_j \hat{p}_j, j \in \{0, 1, ..., n\}$. Otherwise $\hat{h}_i = 0$. $\hat{\mathbf{h}}$ is the one hot vector of the depth level selection results. However, the argmax operation when generating the one hot vector will led to non-differentiable. Therefore, during the backward propagation process, we apply the Gumbel Softmax Strategy and relax the one-hot vector $\hat{\mathbf{h}}$ to $\tilde{\mathbf{h}} = [\tilde{h}_0, \tilde{h}_1, ..., \tilde{h}_n]$ as follows:

$$\tilde{h}_i = \frac{\exp(\hat{p}_i/\tau))}{\sum_{j=0}^{7} \exp(\hat{p}_j/\tau)}, i \in \{0, 1, ..., n\} \tag{2}$$

where τ is the temperature parameter. Using the Gumbel softmax Strategy Jang et al. (2017), we can select the optimal depth-level of octree for machine tasks based on the argmax function during forward propagation process and approximate the gradient of the argmax function by using Eq. (2) in the back propagation process. During the inference stage, we directly select the depth level with the maximum probability in $\mathbf{p}$.

3.4 Training Strategy

Loss Function. In our SPC-Net, we need to train three modules including the octree depth level predictor, the compression module (*i.e.,* the encoder and the decoder) and the task specific network module. As the encoder and the decoder is the same as the VoxelContext-Net Que et al. (2021), we train the compression module based on the same setting as VoxelContext-Net. For the task network module, we train the network based on the same setting as PointNet++ Qi et al. (2017) or VoteNet Qi et al. (2019). For our octree depth level predictor, we train it after the other two modules are pre-trained. When training the octree depth level predictor, we fix the parameters in the compression module and the task network module. And the loss function function used for training our octree depth level predictor is shown here,

$$loss = \sum((\lambda * \mathbf{bpp} + \mathbf{L}) * \tilde{\mathbf{h}}) \tag{3}$$

Our method selects the optimal depth level from n depth levels. $\mathbf{bpp} = (\mathrm{bpp}_1, \mathrm{bpp}_2, ..., \mathrm{bpp}_\mathrm{n})$, bpp means bits per points, which denotes the length of the bit-stream. $\mathrm{bpp_i(i} \in \{0, 1, ..., \mathrm{n}\})$ represents the bpp for constructing the octree at the ith depth levels. We can obtain bpp from the encoder. $\mathbf{L} = (L_1, L_2, ..., L_n)$, and L_i are formulated as follows,

$$L_i = D(f(\hat{x_i}), y_{gt}), i \in \{0, 1, ..., n\} \tag{4}$$

where f is the machine vision task network (*i.e.,* PointNet++ or VoteNet). $\hat{x_i}$ is the reconstructed point cloud from the octree with i depth levels. y_{gt} is the ground true for the machine vision tasks. And D can calcualte the loss between the $f(\hat{x_i})$ and the y_{gt}. $\tilde{\mathbf{h}} = [\tilde{h}_0, \tilde{h}_1, ..., \tilde{h}_n]$, and $\tilde{h}_i$ is designed in equation 2. λ is a hyper-parameter, which is used to balance the trade off of $\mathbf{bpp}$ and $\mathbf{L}$.

4 Experiment

4.1 Dataset

ModelNet. ModelNet Wu et al. (2015) is a widely used benchmark to evaluate the point cloud classification performance, which contains two datasets named ModelNet40 and ModelNet10. ModelNet40 dataset is divided into 40 categories, which has 9843 point cloud data for training and 2468

point cloud data for testing. The ModelNet10 dataset is a subset of the ModelNet40, which only has 10 categories with 3991 point cloudd data for training and 908 point cloud data for testing.

ShapeNet. ShapeNet Yi et al. (2016) contains 16881 point cloud data from 16 object classes. Each point cloud data contains 2-5 parts, with a total of 50 part categures. ShapeNet has 14007 point cloud data for training and 2874 point cloud data for testing.

ScanNet. ScanNet Dai et al. (2017) is a physically available dataset used for the 3D object detection task, which contains 1201 scans for training and 312 scans for testing. Following the VoxelContext-Net Que et al. (2021), we sample 50,000 points from each scan.

4.2 Experiment Details

Baseline. To the best of our knowledge, this is the first point cloud compression framework for both machine vision and human vision tasks. Therefore, we directly use the encoder and the decoder from VoxelContext-Net Que et al. (2021) as our baseline method, which is the start-of-the-art point cloud compression method designed for the human vision task. We also use the same encoder and decoder for point cloud compression in our proposed framework for a fair comparison with the baseline method.

For the baseline methods for the machine vision tasks, we directly use the reconstructed point clouds from VoxelContext-Net as the input of the networks for the machine vision tasks. PointNet++ Qi et al. (2017) is used for the classification task and segmentation task. VoteNet Qi et al. (2019) is adopted for the detection task. For the classification task, we use the octree with the depth levels of 3,4,5 to compress the input raw point clouds. For the segmentation task, we use the octree with the depth levels of 4,5,6 for data compression. For the detection task, we use the octree with the depth levels of 7,8,9 for data compression. As suggested in VoxelContext-Net Que et al. (2021), we train the machine vision task networks with the raw point clouds and evaluate the classification/segmentation/detection results based on the reconstructed point clouds.

Evaluation Metric. We use bit per point (bpp) to denote the bit cost in the compression procedure. For the machine vision tasks, accuracy, mean intersection-over-union (mIoU) and mean average precision (mAP) are used to measure the performance for the classification, segmentation and detection tasks, respectively. For the human vision task, Point-to-Point PSNR Tian et al. (2017) and Chamfer distance (CD) Fan et al. (2017); Huang & Liu (2019) are used to measure the distortion between the reconstructed point cloud and the raw point cloud, which are widely used metric for measuring compression performance.

Implementation Details. We train our model in two stages. At the first stage, we only train the encoder, the decoder and the task specific networks. We use the same training strategy as VoxelContext-Net Que et al. (2021) to train the encoder and the decoder. For different task specific networks (*i.e.*, PointNet++ Qi et al. (2017) and VoteNet Qi et al. (2019)), we follow the settings in their works to train the task specific networks. At the second stage, based on the loss function 3, we train the octree depth predictor by fixing the parameters in the encoder, the decoder and the task specific networks. For the classification task, the hype-parameter λ is set from 0.01 to 16. For the segmentation task, the hype-parameter λ is set from 0.02 to 8. For the detection task, the hype-parameter is set as 0.3, 0.6, 1 and 2.

The whole network is implemented by Pytorch with CUDA support. At the second training stage, we set the batch size as 48. We use the Adam optimizer Kingma & Ba (2015) with the learning rate of 1e-4 for the first 50 epochs, 1e-5 for the next 30 epochs, and 1e-6 for the last 20 epochs.

In our experiments, the maximum depth levels of the octrees are set as 8, 8 and 9 for the human vision task on ModelNet, ShapeNet and ScanNet datasets, respectively, as the predefined depth levels are sufficient for reconstructing high quality point clouds in a highly visual experience. For machine vision, the maximum depth levels of the octree are set as 7, 7 and 9 on ModelNet, ShapeNet and ScanNet datasets, respectively.

4.3 Experiment Results

Classification Task. The classification results of our SPC-Net on the ModelNet10, ModelNet40 and ShapeNet datasets are shown in Figure 3 (a) (b) and (c). It is observed that our proposed

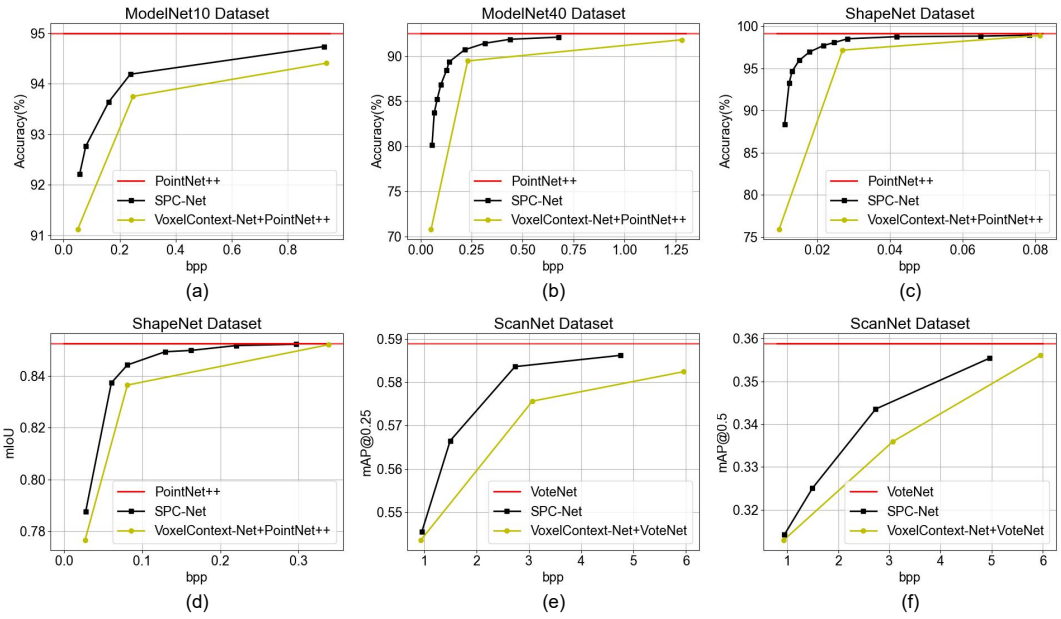

Figure 3: The classification performance of different methods at different bpp on the ModelNet10 (a), ModelNet40 (b) and ShapeNet (c) datasets. And the segmentation performance of different methods at different bpp on the ShapeNet (d). And detection performance of different method at different bpp on the ScanNet datasets (e)(f) .

framework SPC-Net achieves 1% accuracy improvement at 0.05 bpp on the ModelNet10 dataset when compared with our baseline method. On the ModelNet40 dataset, our SPC-Net achieves about 10% accuracy improvement at 0.056 bpp and saves about 0.8 bpp at 91.8% accuracy. On the ShapeNet dataset, our SPC-Net achieves more than 10% accuracy improvement at the 0.01 bpp when compared with our baseline method using 4 octree depth levels. The experimental results demonstrate that our new SPC-Net can improve the performance when the input point cloud is compressed for the classification task.

Segmentation Task. The segmentation results of our SPC-Net on the ShapeNet dataset are shown in Figure 3 (d). We observe that our proposed framework SPC-Net achieves 0.8% mIoU improvement at 0.08 bpp when compared with our baseline method. Our method can save above 10% bpp when the mIoU target is similar when compared with our baseline method. Therefore, our method achieves better performance than the baseline method for the segmentation task.

Detection Task. The detection results of our SPC-Net on the ScanNet dataset are shown in Figure 3 (e) and (f). From Figure 3 (e) we observe that our framework SPC-Net achieves about 0.01 mAP@0.25 improvement at 3 bpp when compared with our baseline method. At the highest bpp, our SPC-Net saves above 20% bpp when compared with our baseline. From figure 3 (f), we observe that our SPC-Net imporves about 0.08 mAP@0.5 and saves about 0.3 bpp when compared with our baseline method at 3 bpp. And at the highest bpp, our SPC-Net saves above 15% bpp when compared with our baseline method. The experimental results demonstrate that our proposed framework can also improve the performance of the detection task.

Human Vision Results. Our SPC-Net achieves exactly the same compression performance as our baseline method VoxelContext-Net Que et al. (2021), (please refer to Appendix A.2 for more details). It should be mentioned that in most 2D images compression for both machine vision and human vision methods Choi & Bajić (2022); Yang et al. (2021); Torfason et al. (2018), further compression performance for human vision always drops in order to achieving better performance for the machine vision tasks. Therefore, this is the advantage that our proposed framework SPC-Net can improve the performance for the machine vision tasks without sacrificing the compression results for human vision.

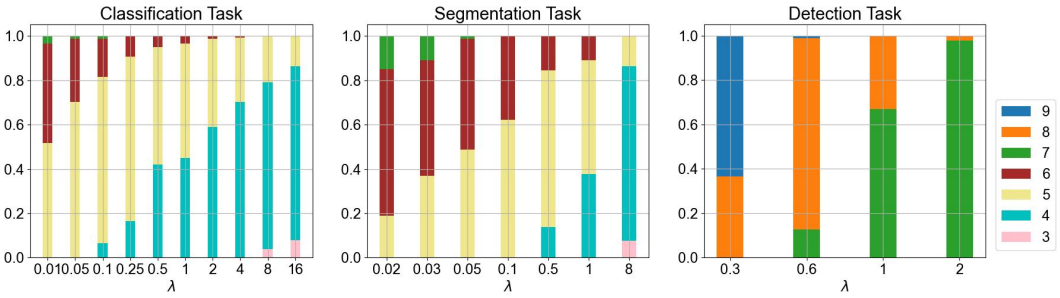

Figure 4: The selection percentage of different depth levels of octree at different λ values of the classification task on the ModelNet40 dataset, the segmentation task on the ShapeNet dataset, and the detection task on the ScanNet dataset. Different colors represents different depth levels.

Table 1: The selection percentage of the octree with different depth levels when our proposed framework SPC-Net is evaluated on the ModelNet40 dataset for classification at $\lambda = 0.25$.

Depth Levels	Simple Categories			General Categories			Complex Categories		
	chair	laptop	bed	monitor	car	airplane	person	curtain	guitar
4	85%	80%	45%	6%	0	0	0	0	0
5	15%	20%	55%	94%	99%	87%	55%	20%	5%
6	0	0	0	0	1%	13%	45%	80%	95%
Accuracy	96%	100%	95%	98%	100%	100%	80%	80%	94%

4.4 Model Analysis

In order to balance the bit-rate cost and the performance of the machine vision tasks in different scenarios, our proposed octree depth level predictor can dynamically adjust the number of the point clouds reconstructed by the octree at different depth levels. The selection percentages at different depth levels of the octrees for different tasks at different λ values are shown in Figure 4. We observe that smaller λ values lead to selection more of higher depth levels. With the increasing of the λ values, our octree depth level predictor will select lower depth levels of the octrees. The selection percentage in Figure 4 demonstrates that our SPC-Net can dynamically select the optimal depth level of the octree at different λ values for different machine vision tasks.

In Table 1, we evaluate our SPC-Net for the classification task on the ModelNet40 dataset when setting $\lambda = 0.25$. It is observed that lower depth levels of the octree are prefered for the simple categories (*e.g.,* chair, laptop, bed) and it can still achieve more than 95% accuracy. Therefore, our SPC-Net will save bits while still achieve promising classification performance. For the "complex" categories (*e.g.,* person, curtain, guitar), our octree depth level predictor prefers higher depth levels. As it is hard to recognize objects from the complex categories, our SPC-Net need to spend more bits for higher depth levels to achieve better classification results. The results demonstrate that our proposed octree depth level predictor can select different depth levels for different input point clouds according to their characteristics (*e.g.,* the relating "simple" or "complex" categories).

4.5 Conclusion

In this work, we have proposed a new scalable point cloud compression framework SPC-Net for both machine vision and human vision tasks. In our SPC-Net, we propose a new scalable bit-stream partitioning method based on the point cloud encoder-decoder structure in order to make the compressed point clouds more suitable for the both tasks. Additionally, considering the purpose of different tasks and the characteristics of different point clouds, we design a new depth level predictor to guide the division of the bit-stream. The experimental results on four benchmark datasets demonstrate that our SPC-Net achieves promising results for three machine vision tasks(*i.e.,* classification, segmentation, detection) without sacrificing the performance of the human vision task.

REFERENCES

Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.

Sourav Biswas, Jerry Liu, Kelvin Wong, Shenlong Wang, and Raquel Urtasun. Muscle: Multi sweep compression of lidar using deep entropy models. *Advances in Neural Information Processing Systems*, 33:22170–22181, 2020.

Sien Chen, Jian Jin, Lili Meng, Weisi Lin, Zhuo Chen, Tsui-Shan Chang, Zhengguang Li, and Huaxiang Zhang. A new image codec paradigm for human and machine uses. *arXiv preprint arXiv:2112.10071*, 2021.

Hyomin Choi and Ivan V Bajić. Scalable image coding for humans and machines. *IEEE Transactions on Image Processing*, 31:2739–2754, 2022.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

Google. Draco 3d graphics compression, 2022. `https://github.com/google/draco`, accessed:2022.

MPEG Group. Mpeg g-pcc tmc13, 2021. `https://github.com/MPEGGroup/mpeg-pcc-tmc13`, accessed:2022.

Lila Huang, Shenlong Wang, Kelvin Wong, Jerry Liu, and Raquel Urtasun. Octsqueeze: Octree-structured entropy model for lidar compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1313–1323, 2020.

Tianxin Huang and Yong Liu. 3d point cloud geometry compression on deep learning. In *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 890–898, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

Nam Le, Honglei Zhang, Francesco Cricri, Ramin Ghaznavi-Youvalari, and Esa Rahtu. Image coding for machines: an end-to-end learned approach. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1590–1594, 2021. doi: 10.1109/ICASSP39728.2021.9414465.

Kang Liu, Dong Liu, Li Li, Ning Yan, and Houqiang Li. Semantics-to-signal scalable image compression with learned revertible representations. *International Journal of Computer Vision*, 129 (9):2605–2621, 2021.

Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9277–9286, 2019.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

Zizheng Que, Guo Lu, and Dong Xu. Voxelcontext-net: An octree based framework for point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6042–6051, 2021.

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

Myungseo Song, Jinyoung Choi, and Bohyung Han. Variable-rate deep image compression through spatially-adaptive feature transform. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2380–2389, 2021.

Dong Tian, Hideaki Ochimizu, Chen Feng, Robert Cohen, and Anthony Vetro. Geometric distortion metrics for point cloud compression. In *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3460–3464. IEEE, 2017.

Robert Torfason, Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Towards image understanding from deep compression without decoding. *OpenReniew. net-ICLR 2018*, 2018.

Jianqiang Wang, Dandan Ding, Zhu Li, and Zhan Ma. Multiscale point cloud geometry compression. In *2021 Data Compression Conference (DCC)*, pp. 73–82. IEEE, 2021a.

Jianqiang Wang, Hao Zhu, Haojie Liu, and Zhan Ma. Lossy point cloud geometry compression via end-to-end learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021b.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Shuai Yang, Yueyu Hu, Wenhan Yang, Ling-Yu Duan, and Jiaying Liu. Towards coding for human and machine vision: Scalable face image coding. *IEEE Transactions on Multimedia*, pp. 1–1, 2021. doi: 10.1109/TMM.2021.3068580.

Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *SIGGRAPH Asia*, 2016.

Wenjie Zhu, Zhan Ma, Yiling Xu, Li Li, and Zhu Li. View-dependent dynamic point cloud compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(2):765–781, 2020.

A APPENDIX

A.1 THE FRAMEWORK

Octree Construction. As shown in Figure 1 (c), octree is a point cloud storage structure, which is beneficial to compression. To construct the octree, we first need to surround the point cloud with the smallest cube. Then the smallest cube will be split into eight sub-cube of the same size. For each sub-cube, if there is no point in the cube, this cube is recorded empty. Otherwise, the cube is recorded nonempty, which means there are some points in this cube. After that, for the nonempty sub-cube, we repeat the above split process to reduce the size of the cube until the depth of the octree reaches the predefined maximum depth value. In the constructed octree, a non-leaf node stands for one cube and the nonempty non-leaf node has eight child nodes that stand for the sub-cubes.

Encoder. The encoder compresses the octree into the bit-stream. All octrees are encoded into the bit-stream from the low depth level to the high depth level, as shown in Figure 1(c). Therefore, we can divide the full bit-stream into two parts according to the selected octree depth.

Decoder and Point Cloud Reconstruction. The decoder will restores the octree from the bit-stream. And the point cloud reconstruction module reconstructs the point cloud coordinates from the octree. The reconstruct point cloud coordinates is the coordinate of the center point of the smallest nonempty cubes. The point cloud coordinates can not only be used in machine vision tasks but also be easily visualized for human vision.

Data Processing. In each octree, all points in one smallest cube will be combined to one point. So the number of points in the reconstructed point cloud will have less number of points than the raw point cloud. Additionally, the reduced number of different point clouds is different, so the output point clouds from the point cloud reconstruction module have different number of points. However, our framework need the size of batch size more than 1 (*e.g.,* 32 or 48). And the point clouds with different number of points can not directly concatenate together to one batch. So we random sample the point cloud based on the predefined number of points to unify the size of the point cloud. As we know, the octree will combine some points to one point. However, each point corresponds to one target for the segmentation task. So if all points in the smallest cube have the same label, we use this label for the new combined point. If the points in one smallest cube have the different label, we use the label of the point which is closest to the combined point.

A.2 HUMAN VISION RESULT

The experimental results of our SPC-Net for human vision are shown in Table 2. In this table, we observe that our SPC-Net achieves the same performance as the VoxelContext-Net.

Table 2: The compression performance of our SPC-Net for human vision on the ModelNet10, ModelNet40, ShapeNet and ScanNet datasets.

Dataset	method	bpp	PSNR	CD
ModelNet10	SPC-Net	6.8508	48.9000	0.003604
	VoxelContext-Net	6.8508	48.9000	0.003604
ModelNet40	SPC-Net	6.4124	48.8835	0.003612
	VoxelContext-Net	6.4124	48.8835	0.003612
ShapeNet	SPC-Net	4.2866	49.1161	0.003528
	VoxelContext-Net	4.2866	49.1161	0.003528
ScanNet	SPC-Net	5.9525	55.2217	0.001740
	VoxelContext-Net	5.9526	55.2236	0.001740

A.3 VISUALIZATION

The visualization results of the segmentation task are shown in Figure 5. From the results of the table and the mug in the first two rows of Figure 5, point clouds reconstructed from the octree with 5 depth levels can achieve similar segmentation performance when compared with the point clouds reconstructed from the octree with more depth levels. Therefore, our octree depth level predictor

prefers the octree with 5 depth levels for the segmentation task in this two cases to save bits. From the results of the car and airplane in the last two rows of Figure 5, the point clouds reconstructed from the octrees with 7 depth levels achieve much better segmentation performance when compared with those octrees with less depth levels. Therefore, our octree depth level predictor selects the octree with 7 depth levels for achieving better segmentation performance in this two cases.

The visualization results of the detection task is shown in Figure 6. In the first row, the point cloud reconstructed from the octree with 7 depth levels achieves the same mAP@0.25 performance when compared with the point clouds reconstructed from the octrees with higher depth levels. Therefore, our octree depth level predictor selects the 7 depth levels in this case to save bits. In the second row, the point cloud reconstructed from the octree with 9 depth levels has much better mAP@0.25 performance than the point cloud reconstructed from the octrees with less depth level. Therefore, our octree depth level predictor selects the octree with 9 depth levels for better detection performance in this case.

It is observed that our proposed octree depth level predictor can select the optimal depth levels of the octrees for different cases, which demonstrate the effectiveness of our proposed octree depth level predictor.

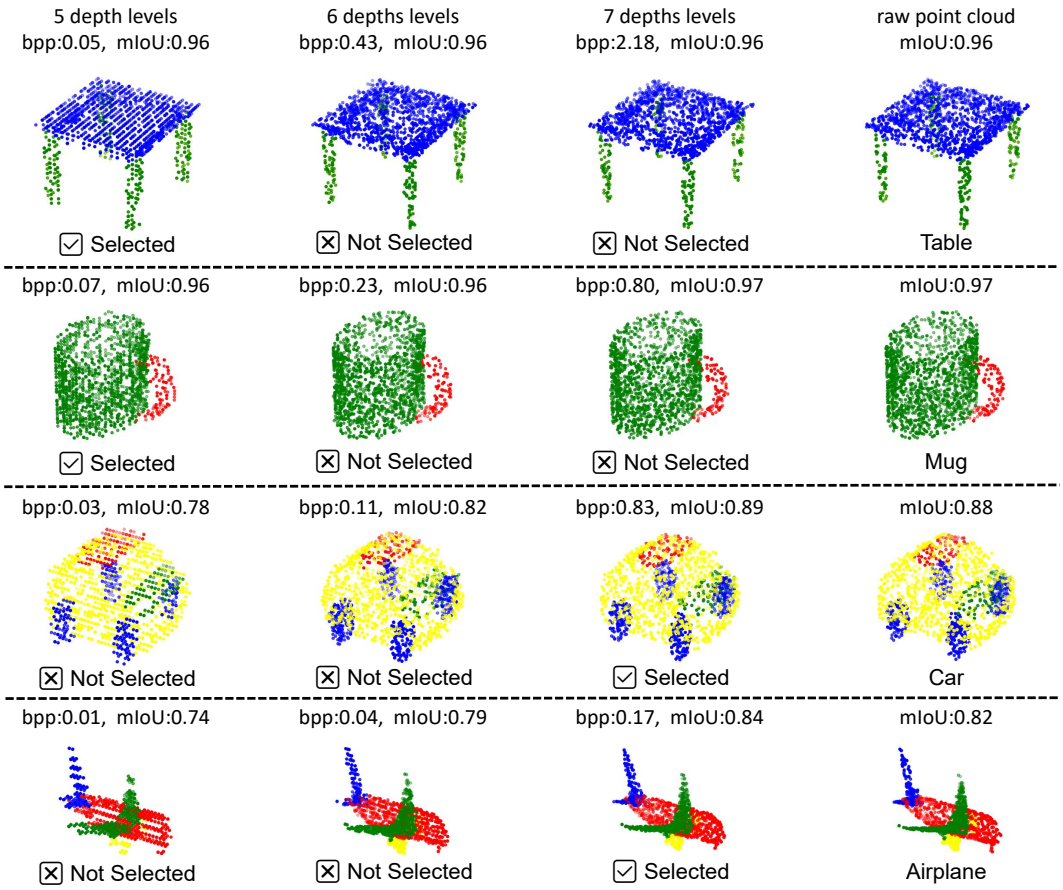

Figure 5: Different qualitative results of segmentation task on ShapeNet dataset."5 depth levels", "6 depth levels" and "7 depth levels" denote that the point cloud is reconstructed by the octree with 5, 6 and 7 depth levels. The λ is predefined as 0.02 in the loss function 3.

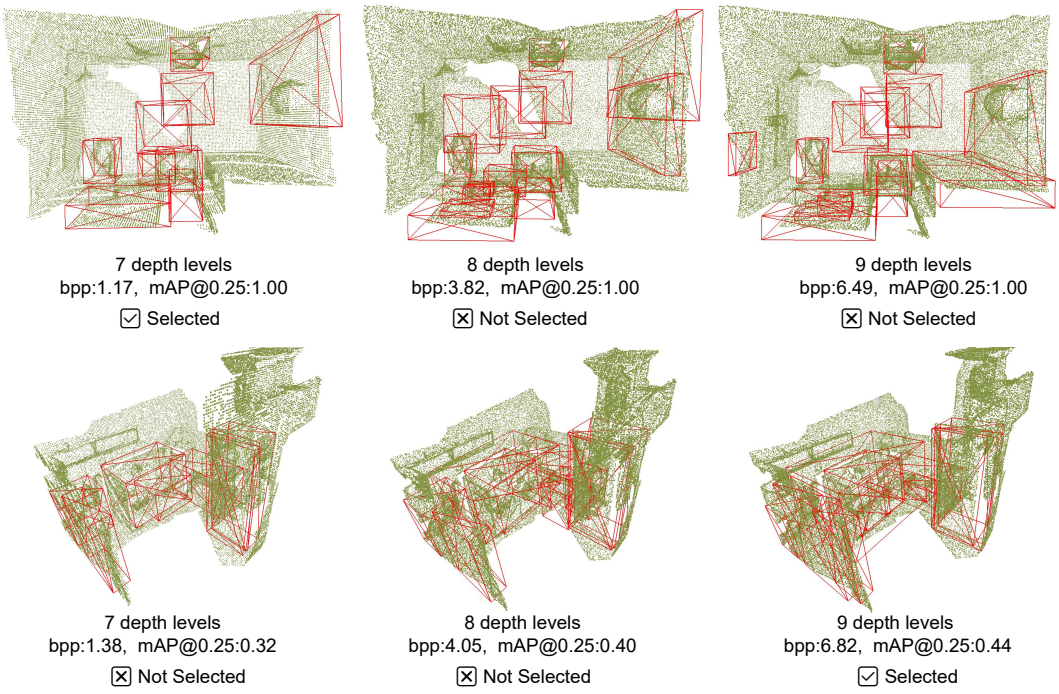

Figure 6: Different qualitative results of detection task on ScanNet dataset. ”7 depth levels”, ”8 depth levels” and ”9 depth levels” mean the point cloud is reconstructed from the octree with 7 depth levels, 8 depth levels and 9 depth levels. The λ predefine as 0.6 in the loss function 3 for the octree depth level predictor to select.