msLPCC: A Multimodal-Driven Scalable Framework for Deep LiDAR Point Cloud Compression

Miaohui Wang[1], Runnan Huang[1], Hengjin Dong[1], Di Lin[2], Yun Song[3], Wuyuan Xie[1*†]

[1] Shenzhen University, Shenzhen 518060, Guangdong, China
[2] College of Intelligence and Computing, Tianjin University, Tianjin 300072
[3] School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha 410004
wang.miaohui@gmail.com, ande.lin1988@gmail.com, sonie@126.com, wuyuan.xie@gmail.com

Abstract

LiDAR sensors are widely used in autonomous driving, and the growing storage and transmission demands have made LiDAR point cloud compression (LPCC) a hot research topic. To address the challenges posed by the large-scale and uneven-distribution (spatial and categorical) of LiDAR point data, this paper presents a new multimodal-driven scalable LPCC framework. For the large-scale challenge, we decouple the original LiDAR data into multi-layer point subsets, compress and transmit each layer separately, so as to ensure the reconstruction quality requirement under different scenarios. For the uneven-distribution challenge, we extract, align, and fuse heterologous feature representations, including point modality with position information, depth modality with spatial distance information, and segmentation modality with category information. Extensive experimental results on the benchmark SemanticKITTI database validate that our method outperforms 14 recent representative LPCC methods.

Introduction

Vehicle to Everything (V2X) plays a vital role in improving the safety and efficiency of transportation systems, which enables vehicles to interact with the external environment, especially in the context of autonomous driving (Lu et al. 2019). Nevertheless, the use of light detection and ranging (LiDAR) point clouds (Sun et al. 2022; Zhao et al. 2022) often results in a large volume of visual data, posing considerable challenges for storage and transmission capabilities in future V2X applications. Consequently, the investigation of a more efficient LiDAR point cloud compression (LPCC) scheme becomes essential to further promote autonomous driving, as depicted in Figure 1.

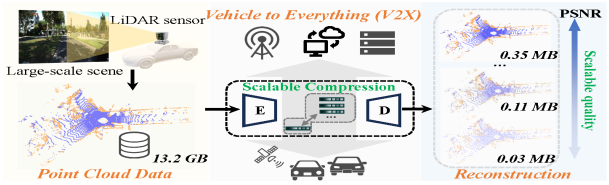

Figure 1: Overview of our scalable Lidar point cloud compression for V2X system. The encoder (E) decouples the original point cloud into scalable features for the V2X system, and the decoder (D) achieves higher reconstruction quality with the increasing number of reconstructed points.

Unlike indoor-scanned and object point clouds, LiDAR point data in autonomous driving is typically acquired in open outdoor scenes (Fang et al. 2022; Zhang et al. 2022b), which exhibit several distinct characteristics: 1) **Multi-modality**. It contains various types of data, such as *raw* points, *depth*, *segmentation*, and *RGB* images. 2) **Large-scale**. It often consists of a significant number of data points (Wang et al. 2022a), which can represent urban streets, building appearances, geographic mapping, *etc*. For example, the *Velodyne HDL-64E LiDAR* sensor can generate nearly 120K points per frame (Geiger, Lenz, and Urtasun 2012). 3) **Uneven distribution**. Due to the physical structure of a LiDAR sensor, the acquired raw points are unevenly distributed in terms of the spatial and category information under the complex environments, as shown in Figure 2. Therefore, it is important to explore a specially-designed LPCC that takes into account these features, as it would be more efficient in compressing LiDAR point data in the context of autonomous driving.

Existing LPCC methods can be broadly divided into non-learning and learning-based compression frameworks: 1) non-learning LPCCs typically involve projecting point clouds into other tractable data formats (Cui et al. 2023), like *depth* and *octree*. However, these methods face limitations when it comes to supporting higher compression ratios for real-time communication and perception, easily leading to the loss of reconstructed points and incorrect inverse

*Corresponding author: Wuyuan Xie
†This work was supported in part by the National Natural Science Foundation of China under Grants 61701310, 62372306 and 2018AAA0102202, in part by the Natural Science Foundation of Guangdong Province under Grants 2023A1515011197 and 2022A1515011245, and in part by the Natural Science Foundation of Shenzhen City under Grant JCYJ20220809160139001.

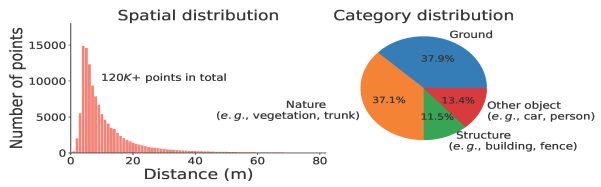

Figure 2: Spatial and category distribution of LiDAR point clouds on the SemanticKITTI database.

mapping. 2) learning-based LPCCs (He et al. 2022) have shown promising results for small-scale point clouds. However, they encounter difficulties when dealing with large-scale sparse LiDAR point data in training latent features and entropy models, where the decrease in compression expression and low efficiency hinders the achievable ultra-low compression ratio. More importantly, existing LPCCs have rarely utilized multimodal learning to further enhance the compression performance.

To improve the performance of LPCC, this paper presents a novel multimodal-driven scalable LPCC framework (msLPCC) that consists of two key modules, namely scalable layer and multimodal encoder, which are specially designed to address the features of large-scale and uneven-distribution of the LiDAR point data respectively. In our msLPCC, the scalable layer decouples the original point clouds into multi-layered subsets with similar geometry shapes but different details, where each subset can globally represent the original point shape, enabling the end-to-end learned codec to achieve a scalable compression paradigm. Moreover, the multimodal-driven encoder improves the learning of the uneven spatial and category distribution of LiDAR point data through the latent feature fusion from depth modality with spatial distance information and segmentation modality with category information. Experimental results on the benchmark SemanticKITTI database (Behley et al. 2019) show that our msLPCC outperforms 14 representative LPCC methods.

The main contributions of msLPCC are as follows:

- To learn efficient compression representations, this paper presents a new multimodal-driven scalable LPCC framework. Through the explicit design of multimodal feature alignment and fusion modules, this method achieves superior coding efficiency on LiDAR point clouds, making it one of the earliest approaches in exploring compression representations.
- To consider the large-scale feature, we propose a solution by decoupling the original point data into multi-layered subsets and compressing each layer separately, which solves the intractable problem of large-scale point clouds and achieves a scalable compression paradigm.
- To take advantage of the uneven spatial and category distributions, we further develop a multimodal encoding scheme that incorporates depth and segmentation features with the point features, which improves the compression efficiency of LiDAR point clouds in an end-to-end manner.

Related Work

According to the compression type, we mainly review some representative LPCC methods from two aspects: 1) format conversion-based methods and 2) direct point-based ones. For a comprehensive overview and performance comparison of point clouds, the reader is referred to (Guo et al. 2021b; Cao et al. 2021).

Format Conversion-based. Existing format conversion-based LPCCs usually convert point clouds into a more compact data structure before compression, including depth image and tree structure. Depth-based methods (Sun et al. 2020; Wang et al. 2022b) map the LiDAR points into a 2D depth image based on a spherical projection. Although the regular 2D depth image format is more tractable, the projection may result in the loss of fine-grained 3D geometric features and incorrect invert projecting at a low bit rate. Tree-based methods use the tree structure to divide and encode point data. Representative tree-based methods include G-PCC (Graziosi et al. 2020) and Draco (Google. 2018), which represent point clouds with *octree* and *kd-tree* respectively. Recently, deep-learned models (Fu et al. 2022; Cui et al. 2023; Huang et al. 2020a) have been developed to learn the probability distribution of octree symbols.

Direct Point-based. Inspired by end-to-end image compression codecs (Hu, Yang, and Liu 2020; Bai et al. 2022; Jiang et al. 2023), direct point-based LPCCs can avoid the problems caused by the data structure conversion. For example, PointNet (Qi et al. 2017a), PointNet++ (Qi et al. 2017b) and Point Cloud Transformer(Guo et al. 2021a) were three benchmark deep network structures for point clouds. Based on these backbones, early point-based compression methods (Yan et al. 2019; Huang and Liu 2019; Gao et al. 2021; Zhang et al. 2022a; Liang and Liang 2022) have been developed to encode the point clouds in an end-to-end manner. These point-based models have shown excellent encoding performance on small-scale object point clouds with a high compression ratio. However, compressing large-scale LiDAR point clouds is challenging, and existing methods often overlook the uneven distribution of spatial and category information. (He et al. 2022; Huang and Wang 2023) introduced a promising point-based compression model for LiDAR point clouds by dividing them into local blocks.

Our msLPCC method is also direct point-based, but we explore a scalable paradigm to decouple the original point clouds into multiple point subsets, thus alleviating the large-scale challenge. Moreover, to represent the uneven spatial and category distributions, we design a multimodal fusion module to reconcile the depth and segmentation features with the point feature. It is worth noting that our work represents one of the earliest attempts to tackle the challenges of LPCC through end-to-end multimodal learning.

Proposed msLPCC Framework

Currently, direct point-based compression methods are severely constrained by the scale and distribution of LiDAR point clouds as illustrated in Figure 2. On the one hand, the large-scale nature of LiDAR point data makes it difficult for point-based models to directly and fully learn efficient com-

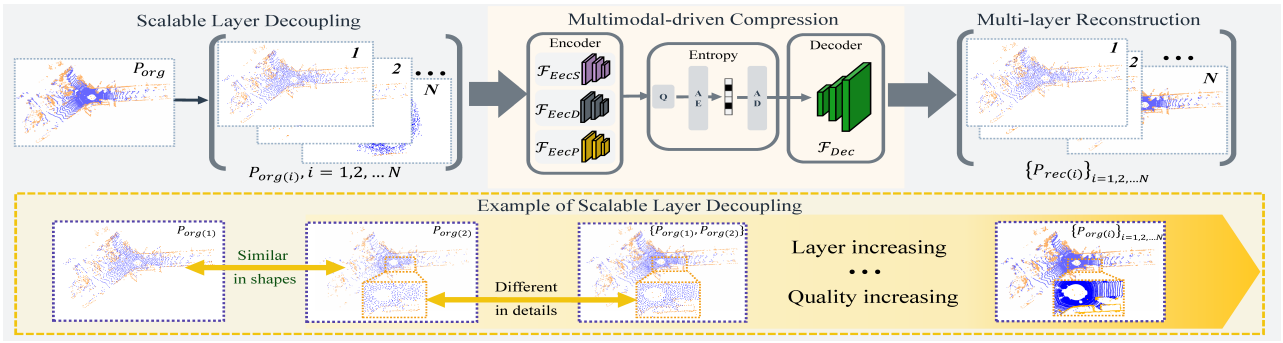

Figure 3: Pipeline of the proposed multimodal-driven scalable compression network. First, the original LiDAR point clouds are decoupled into multi-layered subsets. Then, each layer of points is compressed by an end-to-end multimodal compression network. Finally, each layer of reconstructed point clouds is performed to jointly generate a whole one. The blue box shows a decoupling example of 2-layered point clouds.

pression features. On the other hand, the uneven distribution of LiDAR point data leads to a deep network focusing excessively on dense areas, which seriously affects the quality of end-to-end reconstruction. Based on the above findings, we propose a scalable end-to-end compression framework and describe the details in the following sections.

Overview of msLPCC

The proposed msLPCC framework is mainly composed of five components: scalable layer decoupling, encoder module, entropy module, decoder module, and multi-layer reconstruction, as shown in Figure 3.

Scalable Layer Decoupling. The farthest point sampling (FPS) selects a candidate point by determining the largest distance from a current decoupled layer, which has a good coverage of the original point set P_{org} (Qi et al. 2017b). Specifically, the k-th point p_k in the i-th layer can be obtained by:

$$P^k_{org(i)} = \underset{p_k \in P^*_{org}}{\operatorname{argmax}} \mathcal{D}(p_k, [P_{org(1,\cdots,i-1)}, P^{1,\cdots,k-1}_{org(i)}]), k = 1, 2, ..., |P_{org(i)}|, \quad (1)$$

where $\mathcal{D}(\cdot,\cdot)$ represents the distance between p_k and its nearest neighbor point in the already-selected points of P_{org}. P^*_{org} denotes the rest decoupled points in P_{org}. $|P_{org(i)}|$ denotes the total number of points in the i-th decoupled layer.

As shown at the bottom of Figure 3, P_{org} is decoupled into 2-layer subsets with similar shapes but different details:

$$P_{org} \rightarrow \{P_{org(1)}, \cdots, P_{org(i)}, \cdots, P_{org(N)}\}, \quad (2)$$

where N represents the total number of decoupled layers, which is set as 32 in our experiments.

Encoder Module. msLPCC consists of a depth encoder $\mathcal{F}_{\text{EncD}}$, a segmentation encoder $\mathcal{F}_{\text{EncS}}$, and a point encoder $\mathcal{F}_{\text{EncP}}$. For example, the outputs of these three encoders are aligned and fused to obtain the latent code f_{code} of the i-th layer:

$$f_{code(i)} = \mathcal{F}_{\boldsymbol{\theta}_{\text{a/f}}}\left(f_{depth(i)}, f_{seg(i)}, f_{point(i)}\right), \quad (3)$$

where $\mathcal{F}_{\boldsymbol{\theta}_{\text{a/f}}}$ denotes the alignment and fusion operations. $f_{depth(i)}$, $f_{seg(i)}$, and $f_{point(i)}$ represent the latent features of $\mathcal{F}_{\text{EncD}}$, $\mathcal{F}_{\text{EncS}}$ and $\mathcal{F}_{\text{EncP}}$ at the i-th layer, respectively. It is noted that f_{code} determines the compression ratio.

Entropy Module. The entropy module approximates the quantization by adding uniform noises η in the training phase (Ballé et al. 2018):

$$\hat{f}_{code(i)} = f_{code(i)} + \eta, \quad (4)$$

where $\hat{f}_{code(i)}$ denotes a "quantized" latent code of the i-th layer. In the testing phase, the entropy module directly quantizes $f_{code(i)}$ by rounding, and then uses a trained entropy model and the arithmetic encoder (AE) to generate a bitstream. In the experiments, we use a hyper-prior model (Ballé et al. 2018) as our entropy model for convenience.

Decoder Module. The quantized feature $\hat{f}_{code(i)}$ of the i-th layer will be fed into a decoder to reconstruct the original points $P_{rec(i)}$:

$$P_{rec(i)} = \mathcal{F}_{\text{Dec}}\left(\hat{f}_{code(i)}\right), \quad (5)$$

where $\mathcal{F}_{\text{Dec}}$ represents a decoder module. In the experiments, we adopt a network suggested in (Gao et al. 2021).

Multi-layer Reconstruction. The final point clouds, P_{rec}, are composed of all reconstructed layers:

$$P_{rec} \leftarrow \{P_{rec(1)}, \cdots, P_{rec(i)}, \cdots, P_{rec(N)}\}. \quad (6)$$

It is worth noting that one of the benefits is that our msLPCC supports a scalable reconstruction quality as demonstrated in Figure 3. In other words, we can improve or reduce the reconstruction quality of LiDAR point clouds by increasing or decreasing the number of layers.

Multimodal Point Cloud Compression

In msPLCC, we consider three modalities in LiDAR point clouds, including point cloud modality, depth modality, and segmentation modality. Each modality offers a unique advantage and captures different domain information from a distinct viewpoint (Kumar et al. 2018; Zhao et al. 2021b; Zhou, Zhang, and Foroosh 2021), which can be collected or generated in different ways. The integration of multimodal information aims to compensate for the information

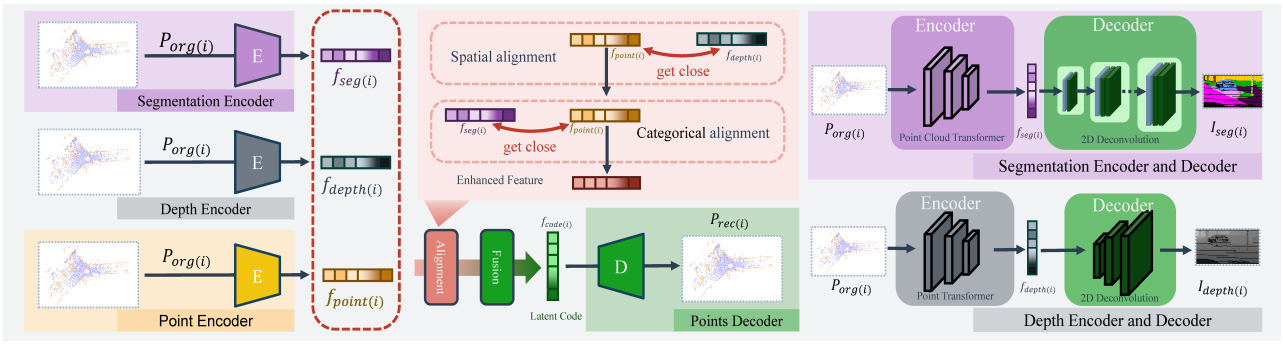

Figure 4: Structure of the proposed multimodal-driven point cloud compression. First, the segmentation and depth encoders are used to extract the segmentation and depth feature representations, respectively. Then, multimodal alignment and fusion are conducted to obtain the latent code feature. The encoder-decoder structures of the segmentation and depth modules are shown on the right-hand side.

loss in each layer resulting from decoupling. In this section, we design an effective multimodal-based compression network that mainly includes multimodal feature representation, alignment, and fusion. The overall pipeline is illustrated in Figure 4.

Problem Formulation. In the compression task, the rate-distortion optimization is used to balance the compression quality and the output bits. In this work, we address the compression problem at an ultra-low compression ratio for real-time communication. In msPLCC, the end-to-end compression of LiDAR point clouds is converted into minimizing the quality loss of the multi-layer reconstruction and the sum of compressed bits per layer. This objective is formulated as:

$$\min \left\{ \mathcal{D}_{quality} \left(P_{org}, \{P_{rec(i)}\}_{i=1,2,\cdots N} \right) + \sum_{i=1}^{N} R\left(\hat{f}_{code(i)} \right) \right\}, \quad (7)$$

where $\mathcal{D}_{quality}$ represents a distortion measurement, and $R_{(\cdot)}$ denotes the related bits of the i-th layer.

Compression Feature Representation. Given the spatial and category unevenness in LiDAR point clouds, we investigate the compression feature representations of segmentation, depth, and point cloud modalities.

(I) Segmentation Feature Representation. LiDAR point clouds usually encompass a mix of large-scale scenes (*e.g.*, road and vegetation) and small-scale objects (*e.g.*, car and traffic signs), highlighting the uneven distribution of categories. To leverage the model to capture these characteristics, it is essential to introduce semantic information, which can provide valuable evidence for object separation (Behley et al. 2019). To achieve this goal, we incorporate the segmentation modality into our msLPCC model.

Specifically, we leverage the Point Cloud Transformer (Guo et al. 2021a) as the encoder $\mathcal{F}_{\text{EncS}}$ to extract segmentation features and adopt a 6-layer deconvolution module as the decoder $\mathcal{F}_{\text{DecS}}$ with a segmentation depth image as the target. It can be formulated as:

$$\begin{aligned} &\min_{\boldsymbol{\theta}_{\text{EncS}}, \boldsymbol{\theta}_{\text{DecS}}} \left\| \mathcal{F}_{\text{DecS}} \left(f_{seg(i)}; \boldsymbol{\theta}_{\text{DecS}} \right) - I_{seg(i)} \right\|_2^2, \\ &s.t. \quad f_{seg(i)} = \mathcal{F}_{\text{EncS}} \left(P_{org(i)}; \boldsymbol{\theta}_{\text{EncS}} \right) \end{aligned} \quad (8)$$

where $\mathcal{F}_{\text{EncS}}$ and $\mathcal{F}_{\text{DecS}}$ denote the segmentation encoder and decoder modules, respectively. $\boldsymbol{\theta}_{\text{EncS}}$ and $\boldsymbol{\theta}_{\text{DecS}}$ denote the model weights of $\mathcal{F}_{\text{EncS}}$ and $\mathcal{F}_{\text{DecS}}$, respectively. $I_{seg(i)}$ denotes the ground-truth of segmentation modality.

(II) Depth Feature Representation. Due to the various application scenarios, LiDAR point clouds exhibit a unique characteristic of being sparse in far distances and dense in near distances. In SemanticKITTI, more than 95% of the LiDAR points are concentrated within a range of 35 meters, with a scanning range of 80 meters in autonomous driving. This characteristic indicates that points at short distances need to be reconstructed with higher fidelity, such as preserving fine-grained shapes.

The depth image obtained from the point cloud spherical projection (Sun et al. 2020) contains a lot of short-distance information, which is expected to help the learning of the spatial localization. Therefore, we employ a typical Point Transformer (Zhao et al. 2021a) as the encoder $\mathcal{F}_{\text{EncD}}$ to extract the depth feature, and adopt a 6-layer deconvolution module (Hu et al. 2021) as the decoder $\mathcal{F}_{\text{DecD}}$ with a depth image $I_{depth(i)}$ as the target. It can be formulated as:

$$\begin{aligned} &\min_{\boldsymbol{\theta}_{\text{EncD}}, \boldsymbol{\theta}_{\text{DecD}}} \left\| \mathcal{F}_{\text{DecD}} \left(f_{depth(i)}; \boldsymbol{\theta}_{\text{DecD}} \right) - I_{depth(i)} \right\|_2^2, \\ &s.t. \quad f_{depth(i)} = \mathcal{F}_{\text{EncD}} \left(P_{org(i)}; \boldsymbol{\theta}_{\text{EncD}} \right) \end{aligned} \quad (9)$$

where $\mathcal{F}_{\text{EncD}}$ and $\mathcal{F}_{\text{DecD}}$ represent the depth encoder and decoder modules, respectively. $\boldsymbol{\theta}_{\text{EncD}}$ and $\boldsymbol{\theta}_{\text{DecD}}$ denote the model weights of $\mathcal{F}_{\text{EncD}}$ and $\mathcal{F}_{\text{DecD}}$, respectively.

(III) Point Feature Representation. The point cloud modality is the original format of LiDAR data, which directly reflects the 3D position relationship. To capture the point cloud features, we train a point-based auto-encoder with Point Cloud transformer (Guo et al. 2021a) as the encoder $\mathcal{F}_{\text{EncP}}$ and a 4-layer 3D-convolution (Gao et al. 2021) as the decoder $\mathcal{F}_{\text{DecP}}$. It can be formulated as:

$$\begin{aligned} &\min_{\boldsymbol{\theta}_{\text{EncP}}, \boldsymbol{\theta}_{\text{DecP}}} \left\| \mathcal{F}_{\text{DecP}} \left(f_{point(i)}; \boldsymbol{\theta}_{\text{DecP}} \right) - P_{org(i)} \right\|_2^2, \\ &s.t. \quad f_{point(i)} = \mathcal{F}_{\text{EncP}} \left(P_{org(i)}; \boldsymbol{\theta}_{\text{EncP}} \right) \end{aligned} \quad (10)$$

where $\mathcal{F}_{\text{EncP}}$ and $\mathcal{F}_{\text{DecP}}$ represent the point encoder and decoder modules, respectively. $\boldsymbol{\theta}_{\text{EncP}}$ and $\boldsymbol{\theta}_{\text{DecP}}$ represent the model weights of $\mathcal{F}_{\text{EncP}}$ and $\mathcal{F}_{\text{DecP}}$, respectively.

Multimodal Feature Alignment. The objective of multimodal feature alignment is to establish the relationship between the point cloud modality, depth modality, and segmentation modality. Since the depth and segmentation modalities contain distinct information, the feature alignment is performed separately for them.

(I) Point-to-Depth. To explore the consistent location information shared between the depth and point cloud modalities, we design a novel point-to-depth contrast $\mathcal{L}_{p\to d}$ which is formulated as:

$$\begin{aligned}\mathcal{L}_{p\to d} = &- \underbrace{\mathop{\mathbb{E}}_{\substack{i=1\\ k\in\{1,\cdots,K\}}}\left[\log\frac{\mathcal{M}_\tau\left(f^k_{point(i)}, f^k_{depth(i)}\right)}{\sum_{k'\in\{1,\cdots,K\}}\mathcal{M}_\tau\left(f^k_{point(i)}, f^{k'}_{depth(i)}\right)}\right]}_{Batch\text{-}level\ Contrast} \\ &- \underbrace{\mathop{\mathbb{E}}_{\substack{i\in\{1,\cdots,N\},\\ k\in\{1,\cdots,K\}}}\left[\log\frac{\mathcal{M}_\tau\left(f^k_{depth(i)}, f^k_{depth(i)}\right)}{\sum_{i'\in\{1,\cdots,N\}}\mathcal{M}_\tau\left(f^k_{depth(i)}, f^k_{depth(i')}\right)}\right]}_{Layer\text{-}level\ Contrast},\end{aligned} \tag{11}$$

where $f^k_{point(i)}$ and $f^k_{depth(i)}$ denote the features for the k-th instance at the i-th layer of point cloud and depth image, respectively. $\mathcal{M}_\tau$ denotes the distance metric with a temperature hyper-parameter τ as suggested in (Tian, Krishnan, and Isola 2020). Eq. (11) is designed based on the following two considerations:

1. The location information is present across various instances. To achieve cross-modal category discrimination, we calculate the metric distance between the point cloud and depth image features obtained from different instances (*e.g.*, the $Batch\text{-}level\ Contrast$ item). The learning goal is to minimize the feature distance for the same instance and maximize it between different instances.
2. Each layer is expected to represent a distinctive location information of the corresponding instance as much as possible. To accomplish this, we adopt an intra-modal instance discrimination by computing the metric distance between the depth image features with different layers from the same instance (*e.g.*, the $Layer\text{-}level\ Contrast$ item). The learning objective is to minimize the feature distance of the same layer and maximize it between different layers.

(II) Point-to-Segmentation. Similarly, to explore the consistent semantic information shared between the segmentation and point cloud modalities, the point-to-segmentation contrast $\mathcal{L}_{p\to s}$ is formulated as:

$$\begin{aligned}\mathcal{L}_{p\to s} = &- \underbrace{\mathop{\mathbb{E}}_{\substack{i=1\\ k\in\{1,\cdots,K\}}}\left[\log\frac{\mathcal{M}_\tau\left(f^k_{point(i)}, f^k_{seg(i)}\right)}{\sum_{k'\in\{1,\cdots,K\}}\mathcal{M}_\tau\left(f^k_{point(i)}, f^{k'}_{seg(i)}\right)}\right]}_{Batch-level\ Contrast} \\ &- \underbrace{\mathop{\mathbb{E}}_{\substack{i\in\{1,\cdots,N\},\\ k\in\{1,\cdots,K\}}}\left[\log\frac{\mathcal{M}_\tau\left(f^k_{seg(i)}, f^k_{seg(i)}\right)}{\sum_{i'\in\{1,\cdots,N\}}\mathcal{M}_\tau\left(f^k_{seg(i)}, f^k_{seg(i')}\right)}\right]}_{Layer-level\ Contrast},\end{aligned} \tag{12}$$

where $f^k_{seg(i)}$ denotes the feature of the i-th layer of point cloud and segmentation modalities for the k-th instance.

Therefore, the learning objective of multimodal feature alignment is:

$$\min_{\boldsymbol{\theta}_{\mathrm{EncP}},\boldsymbol{\theta}_{\mathrm{EncD}},\boldsymbol{\theta}_{\mathrm{EncS}}}(\mathcal{L}_{p\to d}+\mathcal{L}_{p\to s}). \tag{13}$$

Multimodal Feature Fusion. Considering that the spatial location and the semantic category play different roles in information fusion, we propose a two-step multimodal fusion scheme as described below.

(I) Information Fusion. We first conduct an independent multimodal fusion of different feature representations. Intuitively, taking the location information as an example, depth features represent the relative location information while point features represent the absolute location information, and it is important to weigh their contributions to the information fusion accordingly. To achieve this, we adopt an attentive block in (You et al. 2019; Koh et al. 2022), which adaptively fuses the location information between the depth and point features. The fused feature $\hat{f}^k_{d(l)}$ of the location information is obtained through cross-attentive addition, which can be formulated as:

$$\hat{f}^k_{depth(i)} = \sigma\left[\left(\mathbf{W}_{d1}\times f^k_{depth(i)}\right)^T\left(\mathbf{W}_{d2}\times f^k_{depth(i)}\right)\right]\mathbf{W}_{d3}f^k_{point(i)} + f^k_{point(i)}, \tag{14}$$

where $\mathbf{W}_{d1}$, $\mathbf{W}_{d2}$, and $\mathbf{W}_{d3}$ denote the linearly transformations to the location embedding spaces. σ represents the `softmax` function.

Similarly, the fused feature $\hat{f}^k_{s(i)}$ can be formulated as

$$\hat{f}^k_{seg(i)} = \sigma\left[\left(\mathbf{W}_{s1}\times f^k_{seg(i)}\right)^T\left(\mathbf{W}_{s2}\times f^k_{seg(i)}\right)\right]\mathbf{W}_{s3}f^k_{point(i)} + f^k_{point(i)}, \tag{15}$$

where $\mathbf{W}_{s1}$, $\mathbf{W}_{s2}$, and $\mathbf{W}_{s3}$ denote the linearly transformations to the semantic embedding spaces.

(II) Modality Fusion. We first concatenate 256-dimension $\hat{f}^k_{depth(i)}$ and $\hat{f}^k_{seg(i)}$, and then use a single layer of 512-wide full-connection $\mathcal{F}_{\mathrm{Fuse}}$ to fuse them into a 256-dimension feature. We leverage $\mathcal{F}_{\mathrm{DecP}}$ as $\mathcal{F}_{\mathrm{Dec}}$ to decode the reconstructed points. Followed (Yu et al. 2021; Huang et al. 2020b), we adopt $\mathcal{D}_{quality}$ to measure the difference between the reconstructed point and the original point P_{org}. Therefore, the learning objective can be formulated as:

$$\min_{\boldsymbol{\theta}_{\mathrm{Fuse}},\boldsymbol{\theta}_{\mathrm{Dec}}}\mathcal{D}_{quality}\left\{\mathcal{F}_{\mathrm{Dec}}\left[\mathcal{F}_{\mathrm{Fuse}}\left(\hat{f}^k_{depth(i)}\odot\hat{f}^k_{seg(i)};\boldsymbol{\theta}_{\mathrm{Fuse}}\right)+\eta;\boldsymbol{\theta}_{\mathrm{Dec}}\right], P_{org(i)}\right\}, \tag{16}$$

where $\boldsymbol{\theta}_{\mathrm{Fuse}}$ and $\boldsymbol{\theta}_{\mathrm{Dec}}$ represent the model weights of $\mathcal{F}_{\mathrm{Fuse}}$ and $\mathcal{F}_{\mathrm{Dec}}$, respectively. $\odot$ denotes the concatenation.

Experimental Results

In this section, we perform extensive experiments to verify the performance of our msLPCC, providing visual and numeric results. Besides, the ablation experiments are also conducted to demonstrate the effectiveness of the proposed alignment and fusion of different modalities.

Experiment Protocol

Databases. SemanticKITTI (Behley et al. 2019) is a large-scale database captured from driving scenes with a total of 22 sequences. The scene contains $43,552$ scans of Velodyne HDL-64E. The average number of point clouds per scan is over $120K$ points. To fairly evaluate the compression performance, we strictly follow the partition of the SemanticKITTI database, where 10 sequences are used for training and the rest sequences are used for validating and testing as suggested in LPCC (He et al. 2022).

	Method	bpp1	PSNR	bpp2	PSNR	bpp3	PSNR	bpp4	PSNR	BD-Rate Gain
Depth-based	*JPEG XL2020*	0.621	37.96	0.985	42.87	1.384	48.37	1.786	52.40	Anchor
	BPG2018	0.751	37.96	1.038	42.87	1.253	48.37	1.509	52.41	-0.69%
	WebP2023	0.525	37.96	0.740	42.87	0.935	48.37	1.085	52.40	-28.17%
	Wang2022ICRA	0.521	40.95	0.849	50.98	0.958	53.72	1.125	55.40	-39.06%
Tree-based	*Huang2020CVPR*	0.195	48.60	0.534	54.70	1.316	60.66	2.842	66.69	-82.97%
	G-PCC v142021	0.491	58.93	1.202	64.95	2.595	71.00	4.748	77.02	-87.98%
	Cui2023AAAI	0.398	58.95	0.972	65.00	2.141	70.99	3.822	77.02	-89.70%
	Wang2023TPAMI	0.395	58.95	0.939	65.00	2.016	70.99	3.694	77.02	-89.85%
	Fu2022AAAI	0.394	58.95	0.968	65.00	2.101	70.99	3.852	77.02	-90.20%
Point-based	*He2022CVPR*	0.071	34.84	0.113	36.76	0.237	39.01	0.349	42.61	-64.32%
	Qi2017CVPR	0.033	43.40	0.104	44.65	0.203	45.14	0.311	45.24	-88.80%
	Qi2017NeurIPS	0.038	43.98	0.101	44.93	0.237	46.33	0.315	46.57	-89.88%
	Guo2021CVMJ	0.037	44.33	0.092	45.43	0.213	46.59	0.338	47.30	-89.99%
	Zhao2021ICCV	0.039	44.63	0.105	45.42	0.222	46.18	0.291	46.45	-90.23%
	Proposed	**0.031**	**47.81**	**0.115**	**49.28**	**0.215**	**50.07**	**0.312**	**50.34**	**-92.90%**

Table 1: Comparison results of state-of-the-art representative LPCCs. The compression ratio is measured by bpp (bits per point) and the compression quality is measured by PSNR. The BD-Rate gain is calculated using *JPEG XL2020* as the anchor.

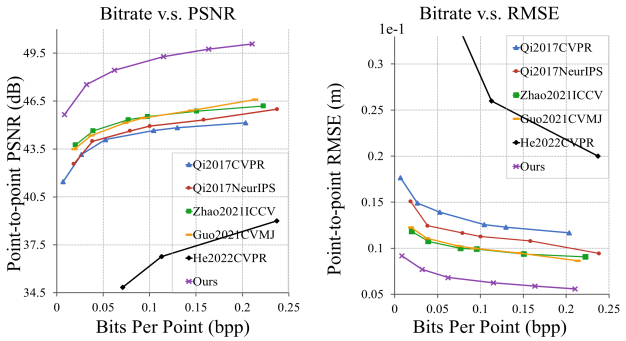

Figure 5: Rate-distortion curves of different point-based methods. The experiments have been conducted on SemanticKITTI in terms of PSNR and RMSE.

Evaluation Metrics. Lossy geometric point cloud evaluation considers both the compression ratio and the reconstruction quality. We use the bits per point (bpp) to demonstrate the compression ratio, and use the widely-used BD-Rate (Cao et al. 2021) to measure the bit rate savings at the equivalent quality. In addition, we use the point-to-point PSNR and RMSE (Schwarz et al. 2018) to evaluate the quality of the reconstructed point clouds.

Comparison Methods. We compare our method with 14 of the latest representative point-based, depth-based, and tree-based LPCC methods. For point-based methods, in addition to *He2022CVPR* (He et al. 2022), we also embed the point-based deep models into our scalable compression framework, including *Qi2017CVPR*'s PointNet (Qi et al. 2017a), *Qi2017NeurIPS*'s PointNet++ (Qi et al. 2017b), *Zhao2021ICCV*'s Point Transformer (Zhao et al. 2021a) and *Guo2021CVMJ*'s Point Cloud Transformer (Guo et al. 2021a) as exiting object point cloud compression methods (Yan et al. 2019; Huang and Liu 2019; Liang and Liang 2022; Zhang et al. 2022a) do. Additionally, our method is compared with depth-based methods including *BPG2018* (Bellard. 2018), *WebP2023* (Zern, Massimino, and Alakuijala 2023), *JPEG XL2020* (Alakuijala et al. 2020), and *Wang2022ICRA* (Wang et al. 2022b), along with tree-based methods including *G-PCC v142021* (Graziosi et al. 2020), *Huang2022CVPR*(Huang et al. 2020a), *Wang2023TPAMI* (sparse tensor) (Wang et al. 2023), *Fu2022AAAI* (Fu et al. 2022), and *Cui2023AAAI* (Cui et al. 2023).

Implementation Details. We decouple the original point clouds into multi-layered subsets. Each scalable layer contains about 2048 points with a total of 32 layers. Our msLPCC is implemented on *PyTorch* and trained by an *NVIDIA Titan RTX* with an *Intel Xeon W-2265*. For the training of a single modality encoder, we set the batch size to 32; while for the training of the msLPPC model, we set the batch size to 16. We use the *Adam* optimizer to train our model with a learning rate of 0.001 for about 200 epochs.

Performance Comparison

The large-scale of LiDAR point clouds exceeds the input capacity of current point-based learning models (Xu et al. 2020). To address this issue, we train other point-based LPCCs using the same scalable framework as our msLPCC. Experimental results show that our scalable compression framework is also effective for other point-based methods. Detailed comparison results are provided below.

Table 1 presents a comprehensive comparison of our msLPCC method with other methods across four different bpp settings (*e.g.*, from ultra-low to high bit rate). For the point-based methods, we compare the compression performance in terms of PSNR at the most similar bpp that can be achieved by the target method. As tree-based and depth-based methods become very inefficient at low bit rates, we also compare these methods in terms of BD-Rate which is computed based on four bpp values. As seen, taking *JPEG XL2020* as the anchor, our msLPCC achieves the best performance. In particular, msLPCC provides 4.92% bit rate savings compared to G-PCC while maintaining similar quality.

Figure 5 shows the comparison of different point-based

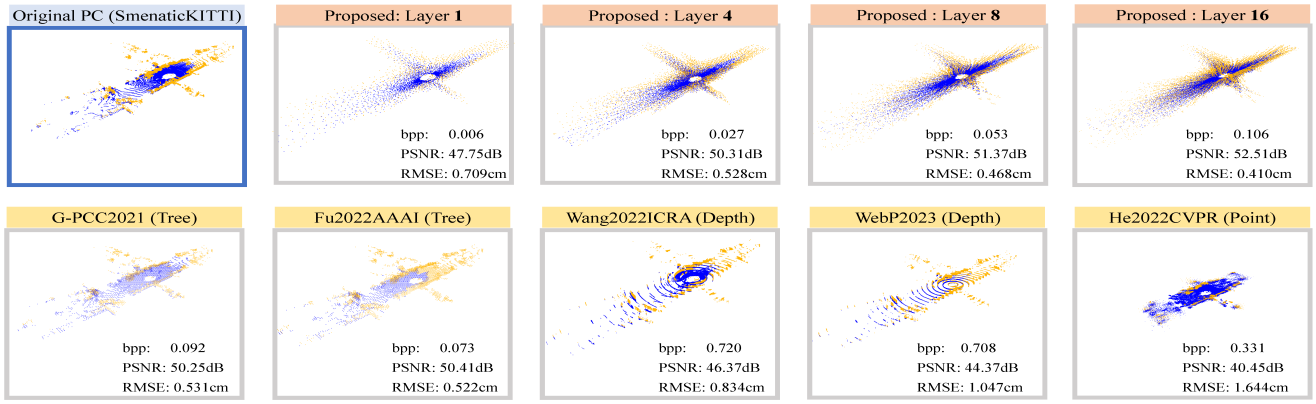

Figure 6: Visualization of our msLPCC and other LPCC methods on the SemanticKITTI database. The first row from left to right: original, Layer 1, Layer 4, Layer 8, and Layer 16. The second row provides some representative tree-based, depth-based, and point-based methods.

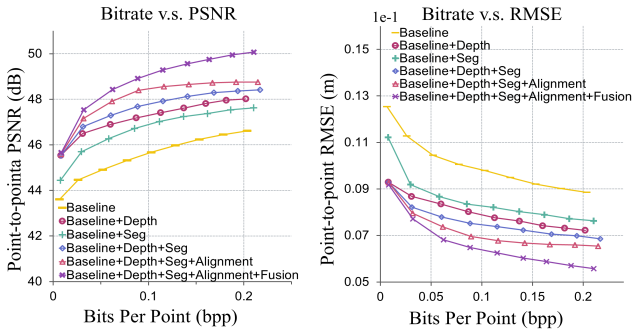

Figure 7: Ablation experiments in terms of rate-distortion curve. The experimental results have been conducted with various multimodal information on SemanticKITTI.

methods in terms of bit rate and distortion. It is evident that msLPCC outperforms other compression models by achieving higher PSNR and lower RMSE at the same bpp.

Figure 6 shows the visualization comparisons of the reconstructed point clouds. To demonstrate the scalability of our msLPCC, we provide the reconstruction point cloud by merging different layers, where Layer "X" represents the reconstruction result of the first X layers. As seen, the quality of the reconstructed points improves as more layers are merged in our msLPCC. In comparison to other methods, our msLPCC effectively preserves the spatial unevenness features of LiDAR point clouds at an ultra-low bit rate.

Ablation Study

To better verify the coding gains of three encoders ($\mathcal{F}_{EncD}$, $\mathcal{F}_{EncS}$, and$\mathcal{F}_{EncP}$) in our msLPCC, we have conducted the additional ablation experiments and the related results are shown in Figure 7, including: (1) "Baseline+Depth+Seg+Alignment+fusion": our complete multimodal compression model; (2) "Baseline+Depth+Seg+Alignment": removing the fusion module and adding aligned features directly; (3) "Baseline+Depth+Seg": removing the alignment module and fusion module, and directly adding the segmentation feature and depth feature to the point feature; (4) "Baseline+Seg": removing the alignment module, fusion module, and depth feature, and directly adding the segmentation feature to the point feature; (5) "Baseline+Depth": removing the alignment module, fusion module, and segmentation feature, and directly adding the depth feature to the point feature; (6) "Baseline": only using $\mathcal{F}_{\text{EncP}}$ as the backbone encoder to obtain the latent code, without using $\mathcal{F}_{\text{EncS}}$ and $\mathcal{F}_{\text{EncD}}$. Figure 7 shows the rate-distortion curves of our model under the conditions (1)-(6). As seen, each component of the multimodal encoder can improve the reconstruction quality in terms of PSNR and reduce the reconstruction error in terms of RMSE.

Conclusion

To address the challenges posed by the large-scale and uneven distribution (both spatially and categorically) of LiDAR point clouds, we propose a new multimodal-driven compression approach. Our multimodal-driven scalable LPCC framework (msLPCC) shows a promising capacity in tackling the distinct characteristics of LiDAR point data. Specifically, our msLPCC decouples the input LiDAR point clouds into multiple layers of point subsets. Each layer undergoes a separate multimodal-driven compression, enabling it to achieve a scalable compression paradigm. To enhance the learning efficiency, we further design special multimodal alignment and fusion module for msLPCC, where the depth, segmentation, and point features are effectively integrated into an end-to-end manner. Experimental results demonstrate the superiority of our method over 14 recent representative compression schemes at the ultra-low bit rate, including tree-based, depth-based, and point-based encoders. We believe that this exploration will improve the efficiency of LPCC and address the increasing storage and transmission demands in autonomous driving and large-scale urban scene reconstruction.

References

Alakuijala, J.; Boukortt, S.; Ebrahimi, T.; Kliuchnikov, E.; Sneyers, J.; Upenik, E.; Vandevenne, L.; Versari, L.; and Wassenberg, J. 2020. Benchmarking JPEG XL image compression. In *SPIE Optics, Photonics and Digital Technologies for Imaging Applications VI*, 187–206.

Bai, Y.; Yang, X.; Liu, X.; Jiang, J.; Wang, Y.; Ji, X.; and Gao, W. 2022. Towards end-to-end image compression and analysis with transformers. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, 104–112.

Ballé, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations(ICLR)*, 1–23.

Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; and Gall, J. 2019. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE International Conference on Computer Vision (ICCV)*, 9297–9307.

Bellard., F. 2018. BPG:Better Portable Graphics. http://bellard.org/bpg, Accessed Sep. 12, 2022.

Cao, C.; Preda, M.; Zakharchenko, V.; Jang, E. S.; and Zaharia, T. 2021. Compression of sparse and dense dynamic point clouds—methods and standards. *Proceedings of the IEEE*, 109(9): 1537–1558.

Cui, M.; Long, J.; Feng, M.; Li, B.; and Kai, H. 2023. OctFormer: Efficient Octree-Based Transformer for Point Cloud Compression with Local Enhancement. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 37, 470–478.

Fang, G.; Hu, Q.; Wang, H.; Xu, Y.; and Guo, Y. 2022. 3DAC: Learning Attribute Compression for Point Clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 14819–14828.

Fu, C.; Li, G.; Song, R.; Gao, W.; and Liu, S. 2022. OctAttention: Octree-Based large-Scale contexts model for point cloud Compression. In *AAAI Conference on Artificial Intelligence (AAAI)*, 625–633.

Gao, L.; Fan, T.; Wan, J.; Xu, Y.; Sun, J.; and Ma, Z. 2021. Point cloud geometry compression via neural graph sampling. In *IEEE International Conference on Image Processing (ICIP)*, 3373–3377.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3354–3361.

Google. 2018. Draco: 3D data compression. https://github.com/google/draco Accessed April 4, 2010.

Graziosi, D.; Nakagami, O.; Kuma, S.; Zaghetto, A.; Suzuki, T.; and Tabatabai, A. 2020. An overview of ongoing point cloud compression standardization activities: Video-based (V-PCC) and geometry-based (G-PCC). *APSIPA Transactions on Signal and Information Processing*, 9(1): 1–17.

Guo, M.-H.; Cai, J.-X.; Liu, Z.-N.; Mu, T.-J.; Martin, R. R.; and Hu, S.-M. 2021a. PCT: Point cloud transformer. *Springer Computational Visual Media*, 7(2): 187–199.

Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; and Bennamoun, M. 2021b. Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12): 4338–4364.

He, Y.; Ren, X.; Tang, D.; Zhang, Y.; Xue, X.; and Fu, Y. 2022. Density-preserving Deep Point Cloud Compression. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2333–2342.

Hu, M.; Wang, S.; Li, B.; Ning, S.; Fan, L.; and Gong, X. 2021. PENet: Towards precise and efficient image guided depth completion. In *IEEE International Conference on Robotics and Automation (ICRA)*, 13656–13662.

Hu, Y.; Yang, W.; and Liu, J. 2020. Coarse-to-fine hyperprior modeling for learned image compression. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 11013–11020.

Huang, L.; Wang, S.; Wong, K.; Liu, J.; and Urtasun, R. 2020a. OctSqueeze: Octree-structured entropy model for LiDAR compression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1313–1323.

Huang, R.; and Wang, M. 2023. Patch-Wise LiDAR Point Cloud Geometry Compression Based on Autoencoder. In *Springer International Conference on Image and Graphics (ICIG)*, 299–310.

Huang, T.; and Liu, Y. 2019. 3D point cloud geometry compression on deep learning. In *ACM International Conference on Multimedia (MM)*, 890–898.

Huang, Z.; Yu, Y.; Xu, J.; Ni, F.; and Le, X. 2020b. Pf-net: Point fractal network for 3d point cloud completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7662–7670.

Jiang, X.; Tan, W.; Tan, T.; Yan, B.; and Shen, L. 2023. Multi-modality deep network for extreme learned image compression. In *AAAI Conference on Artificial Intelligence(AAAI)*, volume 37, 1033–1041.

Koh, J.; Kim, J.; Yoo, J. H.; Kim, Y.; Kum, D.; and Choi, J. W. 2022. Joint 3d object detection and tracking using spatio-temporal representation of camera image and lidar point clouds. In *AAAI Conference on Artificial Intelligence (AAAI)*, 1210–1218.

Kumar, V. R.; Milz, S.; Witt, C.; Simon, M.; Amende, K.; Petzold, J.; Yogamani, S.; and Pech, T. 2018. Near-field depth estimation using monocular fisheye camera: A semi-supervised learning approach using sparse LiDAR data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–3.

Liang, Z.; and Liang, F. 2022. TransPCC: Towards deep point cloud compression via transformers. In *ACM International Conference on Multimedia Retrieval (ICMR)*, 1—-5.

Lu, R.; Zhang, L.; Ni, J.; and Fang, Y. 2019. 5G vehicle-to-everything services: Gearing up for security and privacy. *Proceedings of the IEEE*, 108(2): 373–389.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Conference on Neural Information Processing Systems (NeurIPS)*, 5105—-5114.

Schwarz, S.; Preda, M.; Baroncini, V.; Budagavi, M.; Cesar, P.; Chou, P. A.; Cohen, R. A.; Krivokuća, M.; Lasserre, S.; Li, Z.; et al. 2018. Emerging MPEG standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1): 133–148.

Sun, X.; Wang, M.; Du, J.; Sun, Y.; Cheng, S. S.; and Xie, W. 2022. A Task-Driven Scene-Aware LiDAR Point Cloud Coding Framework for Autonomous Vehicles. *IEEE Transactions on Industrial Informatics*, 1–10.

Sun, X.; Wang, S.; Wang, M.; Cheng, S. S.; and Liu, M. 2020. An advanced LiDAR point cloud sequence coding scheme for autonomous Driving. In *ACM International Conference on Multimedia (MM)*, 2793–2801.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive multiview coding. In *Springer European Conference on Computer Vision (ECCV)*, 776–794.

Wang, J.; Ding, D.; Li, Z.; Feng, X.; Cao, C.; and Ma, Z. 2023. Sparse tensor-based multiscale representation for point cloud geometry compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7): 9055–9071.

Wang, J.; Li, X.; Sullivan, A.; Abbott, L.; and Chen, S. 2022a. PointMotionNet: Point-Wise Motion Learning for Large-Scale LiDAR Point Clouds Sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4419–4428.

Wang, S.; Jiao, J.; Cai, P.; and Wang, L. 2022b. R-PCC: A baseline for range image-based point cloud compression. In *International Conference on Robotics and Automation (ICRA)*, 10055–10061.

Xu, C.; Wu, B.; Wang, Z.; Zhan, W.; Vajda, P.; Keutzer, K.; and Tomizuka, M. 2020. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *Springer European Conference on Computer Vision (ECCV)*, 1–19.

Yan, W.; Liu, S.; Li, T. H.; Li, Z.; and Li, G. 2019. Deep autoencoder-based lossy geometry compression for point clouds. arXiv:1905.03691.

You, H.; Feng, Y.; Zhao, X.; Zou, C.; Ji, R.; and Gao, Y. 2019. PVRNet: Point-view relation neural network for 3D shape recognition. In *AAAI Conference on Artificial Intelligence (AAAI)*, 9119–9126.

Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; and Zhou, J. 2021. Pointr: Diverse point cloud completion with geometry-aware transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 12498–12507.

Zern, J.; Massimino, P.; and Alakuijala, J. 2023. RFC 9403 WebP Image Format. *Animation*, 2: 1–2.

Zhang, J.; Liu, G.; Ding, D.; and Ma, Z. 2022a. Transformer and upsampling-Based point cloud compression. In *ACM International Conference on Multimedia Workshop(ACMMMW)*, 33–39.

Zhang, Y.; Hu, Q.; Xu, G.; Ma, Y.; Wan, J.; and Guo, Y. 2022b. Not all points are equal: Learning highly efficient point-based detectors for 3D LiDAR point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 18953–18962.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021a. Point transformer. In *IEEE international conference on computer vision(ICCV)*, 16259–16268.

Zhao, L.; Ma, K.-K.; Liu, Z.; Yin, Q.; and Chen, J. 2022. Real-Time Scene-Aware LiDAR Point Cloud Compression Using Semantic Prior Representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(8): 5623–5637.

Zhao, S.; Wang, Y.; Li, B.; Wu, B.; Gao, Y.; Xu, P.; Darrell, T.; and Keutzer, K. 2021b. ePointDA: An end-to-end simulation-to-real domain adaptation framework for LiDAR point cloud segmentation. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, 3500–3509.

Zhou, Z.; Zhang, Y.; and Foroosh, H. 2021. Panoptic-polarnet: Proposal-free LiDAR point cloud panoptic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 13194–13203.