Perceive More with Less: LiDAR Point Cloud Compression at Just Recognizable Distortion for 3D Scene Understanding

Miaohui Wang[1], Runnan Huang[1], Taojun Liu[1], Shuyuan Lin[2], Ye Liu[3], Yun Song[4]*

[1]Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University
[2]College of Cyber Security, Jinan University
[3]School of Automation, Nanjing University of Posts and Telecommunications
[4]School of Computer Science and Technology, Changsha University of Science and Technology
wang.miaohui@gmail.com, sonie@126.com

Abstract

Existing LiDAR point cloud (LPC) coding methods primarily focus on balancing compression efficiency and reconstruction quality according to the human vision system (HVS). However, these methods rarely consider the requirements of downstream scene understanding tasks from the perspective of the machine vision system (MVS). To address this challenge, we explore the maximum degree of LPC compression that has negligible impact on perception accuracy, called LPC-based just recognizable compression distortion (*lpcJRCD*). Specifically, we introduce a novel point-wise quantization approach for constructing a MVS-based LiDAR dataset and present a new *lpcJRCD*-guided compression framework tailored for MVS applications. To enhance MVS-based LPC compression efficiency, we develop a dual-feature interaction (DFI) network that fuses point and voxel features. Additionally, we propose a mask-based loss function to ensure accurate point-wise quality level prediction. Experimental results demonstrate the effectiveness of our method in reducing the average bit-rate by up to 94.98% while preserving perception accuracy in autonomous vehicles.

1 Introduction

LiDAR sensing has rapidly developed in recent years, making LiDAR point clouds (LPCs) a fundamental 3D representation for machine vision systems (MVS) in autonomous driving. LPCs enable recognition of both amorphous regions (*e.g.*, vegetation and road) and countable instances (*e.g.*, people and car) [Wu et al. 2021; Teeti et al. 2022; Ma et al. 2024]. However, 3D perception models that rely on LPCs, particularly those performing object recognition and segmentation [Chen et al. 2024], have to process massive amounts of redundant data, resulting in significant storage and transmission overhead. This motivates the urgent need for efficient LPC compression (LPCC) methods specifically tailored to 3D machine perception tasks.

Most LPCC methods aim to balance compression efficiency and reconstruction fidelity from the human-vision system (HVS). Techniques such as octree, range-image projection, and deep latent representations effectively reduce bit-rate [Sun et al. 2023b], and their usefulness in downstream scene understanding has been reported. However, the effect of aggressive compression on 3D machine perception remains underexamined, especially at the limit of tolerable distortion.

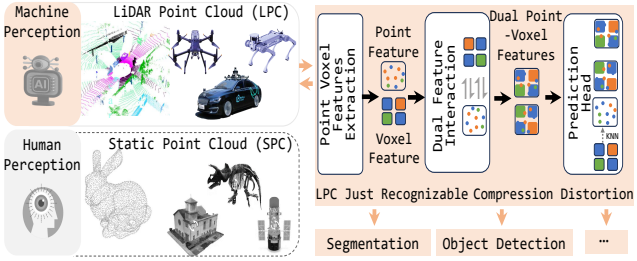

Figure 1: Machine vision system (MVS)-based LiDAR point cloud compression (LPCC) for autonomous vehicles.

To maintain the accuracy of MVS-based systems, lossless or near-lossless compression preserves accuracy but incurs impractically high bit-rates. This motivates the need for an LPCC framework that directly balances bit-rate and perception accuracy. A key requirement is to characterize how compression artifacts influence semantic understanding and to define the **just recognizable compression distortion** (JRCD): the highest distortion that still sustains acceptable machine-perception accuracy. An overview of our MVS-based LPCC pipeline is shown in Figure 1.

Although several MVS-based methods have been explored [Wang et al. 2025a; Xie et al. 2024b; Zhang et al. 2021], they primarily target 2D tasks such as classification and detection. Recent extensions to static point clouds (SPCs) from *ShapeNet* [Liu, Hu, and Zhang 2023] and *ScanNet* [Xie et al. 2024a] achieve progress but remain limited:

1) SPC datasets are not originally designed for machine-perception tasks and contain relatively simple geometric and semantic distributions;
2) Evaluations are restricted to basic tasks (*e.g.*, classification or detection) while overlooking complex semantic understanding tasks such as segmentation;
3) Existing methods have not systematically addressed

*Corresponding author: Yun Song

MVS-based compression under the JRCD condition.

Moreover, SPCs differ significantly from LPCs in their statistical properties [Wang et al. 2025b], indicating that previous MVS-based compression methods have not addressed the complexity of dynamically acquired LiDAR data in autonomous driving. To make matters worse, no public MVS-oriented LPCC dataset currently exists, hindering the development and evaluation of LPC communication frameworks.

To address aforementioned issues, we propose a novel MVS-based LPC compression framework (*lpcJRCD*), tailored for semantic scene understanding in autonomous driving. The key contributions of this work are:

- A novel point-wise quantization strategy that allows point clouds with different quality levels (QLs) to be encoded in a shared octree structure, avoiding excessive bit-rate overhead. Based on this, we construct one of the earliest and largest MVS-oriented LPCC datasets, consisting of 4,071 annotated scenes, over 100,000 objects and their point-wise quantization labels.
- A dual feature interaction (DFI) module that effectively fuses point-wise and voxel-wise features, coupled with a new mask-based loss to enhance the accuracy of predicted QLs, as shown in Figure 5.
- An end-to-end *lpcJRCD*-guided LPCC framework that predicts optimal QLs to guide point-wise compression under the JRCD condition. To our knowledge, this is among the first efforts that explicitly target MVS-based LPCC for perception-critical applications.

Extensive experiments on the *SemanticKITTI* [Behley et al. 2019], *nuScenes* [Caesar et al. 2020], and *KITTI* [Geiger, Lenz, and Urtasun 2012] benchmarks demonstrate that our *lpcJRCD* achieves significantly better compression while maintaining high machine perception accuracy.

2 Related Work

In this section, we briefly review representative HVS-based 3D LPC and early MVS-based image coding methods.

2.1 HVS-based 3D LPC Compression

HVS-based LPCCs aim to reduce the size of point clouds while preserving its geometric accuracy and features. Existing methods mainly include octree-based, image-based, and autoencoder-based LPCC methods. 1) Octree-based methods encode quantized points using an octree and then perform entropy coding via handcrafted methods [Sun et al. 2023a; Yu et al. 2023] or learning-based methods [Huang et al. 2020; Que, Lu, and Xu 2021; Chen et al. 2022; Fu et al. 2022; Song et al. 2023; Fan et al. 2024; Cui et al. 2023; Wang et al. 2023]. 2) Image-based methods [Zhou et al. 2022; Xue et al. 2024] generally project 3D point clouds into 2D range images, which are then compressed using traditional image or video compression schemes [Wang, Ngan, and Li 2016]. 3) Autoencoder-based methods [Huang and Liu 2019; You and Gao 2021; Wiesmann et al. 2021; Wang et al. 2024] typically learn information-intensive latent features via deep networks to compress LPCs.

While these HVS-based LPCCs have shown effectiveness for user experience-based applications, they have only demonstrated promising balance between visual quality and compression efficiency. However, these methods rarely analyze the quantitative impact of the maximum compression level on LPCs from the perspective of machine perception.

2.2 MVS-based 2D and 3D Data Compression

2D Image Compression. MVS-based 2D image compression refers to the maximum coding distortion with negligible effects on recognition accuracy. For example, Jin *et al.* [Jin et al. 2021] first explored a deep machine vision model and demonstrated that images have specific quantization parameters for deep classification models. Zhang *et al.* [Zhang et al. 2021] analyzed the impact of quantization parameters for video compression in *classification* and *detection* tasks. Furthermore, Zhang *et al.* [Zhang et al. 2023] constructed an MVS-based image compression dataset with an expanded number of object categories and proposed an effective two-step quantization prediction model.

3D SPC Compression. Several MVS-based 3D compression methods have been developed for SPC data (see Fig. 1), aiming to optimize both human and machine vision performance. Liu *et al.* [Liu, Hu, and Zhang 2023] proposed a dual-branch model combining point selection with octree-based compression (PCHM-Net), which achieves competitive classification accuracy on SPC benchmarks such as *ModelNet*, *ShapeNet*, and *ScanNet*. Xie *et al.* [Xie et al. 2024a] introduced an ROI-guided geometry compression model that employs ROI-supervised residual refinement to preserve semantic details, improving detection accuracy on *ScanNet* and *SUN RGB-D*. Later, Xie *et al.* [Xie et al. 2024b] further proposed a scalable point cloud geometry compression model leveraging a task-guided optimization to enhance perception accuracy without sacrificing compression efficiency on *ModelNet* and *ShapeNet*.

As far as we know, there is no MVS-oriented compression dataset and corresponding compression method for 3D LPC data. Given the significant differences between 3D LPC and 2D image or 3D SPC data, we hence establish a new *lpcJRCD* dataset and propose a new *lpcJRCD*-guided LPCC framework, promoting the widespread application of LPC data in autonomous driving.

3 Proposed MVS-based Framework

In autonomous driving, MVS-based LPCC aims to achieve the maximum compression ratio that has a negligible impact on perception accuracy. In other words, the original and compressed LPCs satisfy the following relationship:

$$|\mathcal{M}(\mathcal{F}_{\mathrm{mvs}}(\mathbf{P}_{\mathrm{org}}), \mathbf{P}_{\mathrm{gt}}) - \mathcal{M}(\mathcal{F}_{\mathrm{mvs}}(\mathcal{F}_{\mathrm{codec}}(\mathbf{P}_{\mathrm{org}}, \mathbf{Q})), \mathbf{P}_{\mathrm{gt}})| < T, \quad (1)$$

where $\mathcal{M}$ denotes a perception quality measurement, $\mathbf{P}_{\mathrm{org}}$ denotes the input original LPC, $\mathbf{P}_{\mathrm{gt}}$ represents the ground-truth perception label, $\mathcal{F}_{\mathrm{codec}}$ represents an LPC codec, $\mathcal{F}_{\mathrm{mvs}}$ represents a perception model, $\mathbf{Q}$ represents a quality level set used for $\mathbf{P}_{\mathrm{org}}$, and T denotes an *lpcJRCD* threshold used to notify the affordable performance degradation in terms of perception accuracy.

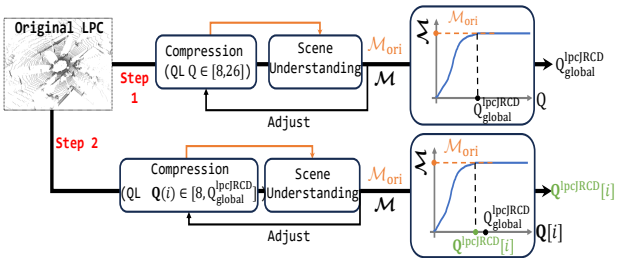

Figure 2: Pipeline of our JRCD dataset construction.

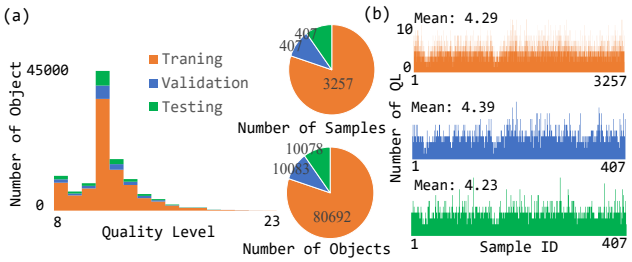

Figure 3: Statistics of our proposed JRCD dataset.

The *lpcJRCD* module is employed to guide $\mathcal{F}_{\text{codec}}$ to achieve lower bit-rate without significantly affecting downstream perception accuracy as well as avoiding heavy computation for obtaining optimum QL values. Although traditional schemes can iteratively search for the optimal QL set for $\mathbf{P}_{\text{org}}$, it is inefficient and impractical for industry applications. Therefore, we propose a deep learned *lpcJRCD* model to predict the optimal QL, and it is formulated as:

$$\mathbf{Q}^{*} = \mathcal{F}_{\text{lpcJRCD}}(\mathbf{P}_{\text{org}}). \tag{2}$$

3.1 Proposed LiDAR-based JRCD Dataset

Raw LPC Data. To ensure universality and openness, we directly construct our MVS-based LPC dataset based on the widely-used *SemanticKITTI* dataset in autonomous vehicles. *SemanticKITTI* includes 20 classes: 8 countable classes and and 12 uncountable classes. Points in countable classes are also assigned an identity, and there are 4071 samples selected as our raw LPC data.

Perception and Compression. We employ a mask-based panoptic segmentation model [Marcuzzi et al. 2023] as our machine perception model $\mathcal{F}_{\text{mvs}}$ in Eq. (1). Assuming that there are a total of $|C|$ classes in $\mathbf{P}_{\text{org}}$, the metric [Milioto et al. 2020] used to evaluate the segmentation and recognition performance is then defined as:

$$\mathcal{M}(\mathbf{P}_{\text{pred}}, \mathbf{P}_{\text{gt}}) = \frac{1}{|C|}\sum_{c=1}^{|C|}(\underbrace{\mathcal{M}_{\text{mIoU}}(\mathcal{F}^{c}_{\text{tp}}(\mathbf{P}_{\text{pred}}, \mathbf{P}_{\text{gt}}))}_{\text{segmentation quality (SQ)}} \times \underbrace{\frac{|\mathcal{F}^{c}_{\text{tp}}(\mathbf{P}_{\text{pred}}, \mathbf{P}_{\text{gt}})|}{|\mathcal{F}^{c}_{\text{tp}}(\mathbf{P}_{\text{pred}}, \mathbf{P}_{\text{gt}})| + 0.5|\mathcal{F}^{c}_{\text{fp}}(\mathbf{P}_{\text{pred}}, \mathbf{P}_{\text{gt}})| + 0.5|\mathcal{F}^{c}_{\text{fn}}(\mathbf{P}_{\text{pred}}, \mathbf{P}_{\text{gt}})|}}_{\text{recognition quality (RQ)}}), \tag{3}$$

where $\mathbf{P}_{\text{pred}}$ and $\mathbf{P}_{\text{gt}}$ represent the predicted and ground-truth segmentation results of $\mathbf{P}_{\text{org}}$, respectively. $\mathcal{F}^{c}_{\text{tp}}$, $\mathcal{F}^{c}_{\text{fn}}$, and $\mathcal{F}^{c}_{\text{fp}}$ determine the true positive set, false negative set, and false positive set of the class c, respectively.

Meanwhile, we use the widely-adopted octree-based compression method as $\mathcal{F}_{\text{codec}}$, which is roughly divided into quantization $\mathcal{F}_{\text{qua}}$, octree compression $\mathcal{F}_{\text{oct}}$, and inverse quantization $\mathcal{F}_{\text{iqua}}$:

$$\mathcal{F}_{\text{codec}}(\mathbf{P}_{\text{org}}, \mathbf{Q}) = \mathcal{F}_{\text{iqua}}(\mathcal{F}_{\text{oct}}(\mathcal{F}_{\text{qua}}(\mathbf{P}_{\text{org}}, \mathbf{Q}))), \tag{4}$$

where $\mathcal{F}_{\text{qua}}$ is the only source that generates compression distortion, and $\mathcal{F}_{\text{oct}}$ is lossless. Then, the relationship between the reconstructed and original LPCs is simplified as:

$$\bar{\mathbf{P}}_{\text{rec}} = \mathcal{F}_{\text{iqua}}(\mathcal{F}_{\text{qua}}(\mathbf{P}_{\text{org}}, \mathbf{Q})), \tag{5}$$

where $\bar{\mathbf{P}}_{\text{rec}}$ represents the reconstruction of $\mathbf{P}_{\text{org}}$.

Label Annotation. We aim to utilize the aforementioned perception ($\mathcal{F}_{\text{mvs}}$) and compression ($\mathcal{F}_{\text{codec}}$) models to annotate every point in $\mathbf{P}_{\text{org}}$. However, given that $\mathbf{P}_{\text{org}}$ contains over 100K points, annotating each point individually is impractical. Additionally, since the MVS-based model $\mathcal{F}_{\text{mvs}}$ is commonly tailored for object of interest (OOI), it is reasonable to assume that all points within an object share the same QL label. Therefore, we annotate the QL label at the object level without degrading general performance. Notably, previous work treats each instance to be detected as an OOI, while the definition of 'OOI' in our *lpcJRCD* dataset is more general, considering countable instances and other uncountable classes as the OOI.

As shown in Figure 2, we firstly compress 4071 original $\mathbf{P}_{\text{org}}$ using the QL value from 8 to 26 to obtain encoded samples with different distortion levels. Then, we find the minimum QL value that satisfies Eq. (1) as our global *lpcJRCD* label in the experiments:

$$\begin{aligned} &\mathrm{Q}^{\text{lpcJRCD}}_{\text{global}} = \min_{8\le \mathrm{Q}\le 26} \mathrm{Q} \\ &\text{s.t. } |\mathcal{M}(\mathcal{F}_{\text{mvs}}(\mathbf{P}_{\text{org}}), \mathbf{P}_{\text{gt}}) - \mathcal{M}(\mathcal{F}_{\text{mvs}}(\bar{\mathbf{P}}_{\text{rec}}), \mathbf{P}_{\text{gt}})| < T. \end{aligned} \tag{6}$$

where T is empirically set to 0.1% that is the retained precision of existing segmentation models. Meanwhile, it is worth noting that some compressed $\bar{\mathbf{P}}_{\text{rec}}$ may exhibit higher perception accuracy compared to the original $\mathbf{P}_{\text{org}}$. One of the main reasons is that $\mathcal{F}_{\text{codec}}$ may filter out some high-frequency noises which can be helpful for subsequent segmentation or recognition task. It is noted that this phenomenon has also been observed in 2D image classification and object detection [Zhang et al. 2021, 2023].

After determining the global $\mathrm{Q}^{\text{lpcJRCD}}_{\text{global}}$, we further obtain point-level QLs for the i-th point: $\mathbf{Q}^{\text{lpcJRCD}}[i] \le \mathrm{Q}^{\text{lpcJRCD}}_{\text{global}}$. Specifically, we sort the objects in $\mathbf{P}_{\text{org}}$ in the descending order of the number of points, and then reduce their QL value $\mathbf{Q}[i]$ sequentially to find the minimum QL that can maintain perceptual accuracy as defined in Eq. (6).

Dataset Statistics. The proposed MVS-based LPC dataset is divided into training, validation, and testing sets in the ratio of 8:1:1. The training set contains 3257 samples and 80692 objects, the validation set contains 407 samples and 10083 objects, and the testing set contains 407 samples and 10078 objects. Figure 3 (a) shows the distribution of our MVS-based LPC dataset in the three sets. Figure 3 (b) shows the number of QLs in each sample. As seen, most of the objects

Algorithm 1 Proposed point-wise quantization $\mathcal{F}_{\text{qua}}$

Input: The *float32*-type original LPC, $\mathbf{P}_{\text{org}}$. The point-wise QLs, $\mathbf{Q}$. The offset of the LPC to the origin, μ.
Output: The quantized LPC $\bar{\mathbf{P}}_{\text{qua}}$

Quality level list: $\mathbf{L} = \texttt{unique}(\mathbf{Q})$
Convert $\mathbf{P}_{\text{org}}$ into double type.
Initialize a temporary reconstruction $\hat{\mathbf{P}}_{\text{rec}} = \{\}$
for k in $(1, 2, \cdots, |\mathbf{L}|)$ **do**
  Points with QL $\mathbf{L}[k]$: $\mathbf{P}^k_{\text{org}} = \{\mathbf{P}_{\text{org}}[i] \mid \mathbf{Q}[i] == \mathbf{L}[k]\}_{i=1}^{N}$
  Quantization: $\bar{\mathbf{P}}^k_{\text{qua}} = \texttt{round}(\frac{\mathbf{P}-\mu}{\max(|\mathbf{P}-\mu|)} \cdot 2^{\mathbf{L}[k]})$,
  Inverse quantization: $\hat{\mathbf{P}}^k_{\text{rec}} = \hat{\mathbf{P}}^k_{\text{qua}} \cdot \frac{\max(|\mathbf{P}-\mu|)}{2} + \mu$.
  Append $\hat{\mathbf{P}}^k_{\text{rec}}$ to $\hat{\mathbf{P}}_{\text{rec}}$: $\hat{\mathbf{P}}_{\text{rec}} = \{\hat{\mathbf{P}}^1_{\text{rec}}, \cdots, \hat{\mathbf{P}}^k_{\text{rec}}\}$
end for
Point cloud quantization: $\bar{\mathbf{P}}_{\text{qua}} = \texttt{round}(\frac{\hat{\mathbf{P}}-\mu}{\max(|\mathbf{P}-\mu|)} \cdot 2^{\max(\mathbf{L})})$.

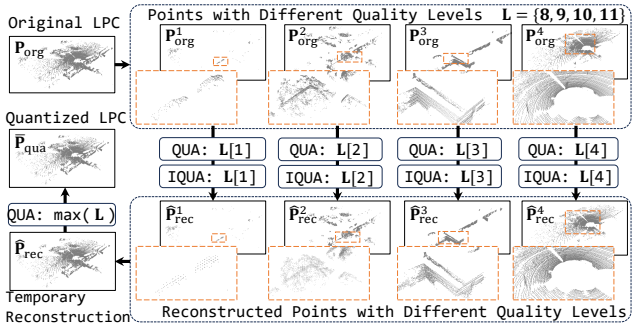

Figure 4: An example of the proposed point-wise quantization algorithm. QUA denotes quantization, IQUA denotes inverse quantization, and QL denotes quality level.

are around 11, most labels contain two or more QLs, and the mean number of QLs is about four.

3.2 Proposed Point-wise Quantization

The quantization $\mathcal{F}_{\text{qua}}$ can be generally formulated as:

$$\bar{\mathbf{P}}_{\text{qua}} = \mathcal{F}_{\text{qua}}(\mathbf{P}_{\text{org}}, \mathbf{Q}) = \texttt{round}(\frac{\mathbf{P}_{\text{org}} - \mu}{s(\mathbf{Q})}), \tag{7}$$

where $\mu = (\min(\mathbf{P}^x_{\text{org}}), \min(\mathbf{P}^y_{\text{org}}), \min(\mathbf{P}^z_{\text{org}}))$ denotes an offset of $\mathbf{P}_{\text{org}}$ from the origin, and s represents a quantization step. The inverse quantization $\mathcal{F}_{\text{iqua}}$ is formulated as:

$$\bar{\mathbf{P}}_{\text{rec}} = \mathcal{F}_{\text{iqua}}(\bar{\mathbf{P}}_{\text{qua}}) = \bar{\mathbf{P}}_{\text{qua}} \cdot s + \mu. \tag{8}$$

However, there are two issues when directly using Eq. (7) to quantize points in $\mathbf{P}_{\text{org}}$: 1) $\mathbf{P}_{\text{org}}$ encoded with different quantization steps needs additional QL information to ensure reconstruction accuracy. 2) The octree divides the x-, y-, and z-axis in 3D space using a binary division, and the spatial range that an octree with a depth of n can represent is from 0 to 2^n. Octree nodes from larger quantization steps s cannot serve as parent nodes for those from smaller s, and this requires constructing separate octrees for each quantization step s, causing *overhead bits explosion*.

To address the above issues, $\mathcal{F}_{\text{qua}}$ should satisfy the following two conditions:

- **Condition 1:** There should be a divisible relationship between adjacent quantization steps. This condition ensures that $\mathbf{P}_{\text{org}}$ quantized with a larger quantization step s will not lose accuracy after further quantization with a smaller s. The associated proof is provided as follows:

Proof of *Condition 1*:
Assuming the quantization step $s_1 > s_2$.
$\bar{\mathbf{P}}_{\text{qua}} = \texttt{round}(\frac{\mathbf{P}-\mu}{s})$,
$\bar{\mathbf{P}}_{\text{rec}} = \texttt{round}(\frac{\mathbf{P}-\mu}{s}) \cdot s_1 + \mu$
$\bar{\mathbf{P}}'_{\text{qua}} = \texttt{round}(\frac{\bar{\mathbf{P}}-\mu}{s}) = \texttt{round}(\frac{\mathbf{P}-\mu}{s}) \cdot \texttt{round}(\frac{s}{s})$
$\bar{\mathbf{P}}'_{\text{rec}} = \texttt{round}(\frac{\mathbf{P}-\mu}{s}) \cdot \texttt{round}(\frac{s}{s}) \cdot s_2 + \mu$
$\bar{\mathbf{P}}_{\text{rec}} = \bar{\mathbf{P}}'_{\text{rec}} \rightarrow \texttt{round}(\frac{s}{s}) = \frac{s}{s}$

- **Condition 2:** There should be a $2\times$ relationship between adjacent quantization steps to ensure that the relative positions of points in $\mathbf{P}_{\text{org}}$ represented by different octrees at different depths remain unchanged. The associated proof is provided as follows:

Proof of *Condition 2*:
Under the condition that $\texttt{round}(\frac{s}{s}) = \frac{s}{s}$, $\frac{s}{s} \geq 2$.
$\bar{\mathbf{P}}_{\text{qua}} = \texttt{round}(\frac{\mathbf{P}-\mu}{s})$, $\bar{\mathbf{P}}'_{\text{qua}} = \texttt{round}(\frac{\bar{\mathbf{P}}-\mu}{s}) \cdot \frac{s}{s}$
Assuming the depth of octree for $\bar{\mathbf{P}}_{\text{qua}}$ and $\bar{\mathbf{P}}'_{\text{qua}}$ is n_1 and n_2.
$\bar{\mathbf{P}}_{\text{qua}}/2^n = \bar{\mathbf{P}}'_{\text{qua}}/2^n \rightarrow \frac{s}{s} = 2^{n-n}$

To satisfy the above two conditions, we propose a point-wise quantization method to preserve the $\mathbf{Q}[i]$-bit accuracy of the i-th point $\mathbf{P}[i]$ as follows:

$$\bar{\mathbf{P}}_{\text{qua}}[i] = \texttt{round}(\frac{\mathbf{P}_{\text{org}}[i] - \mu}{\max(|\mathbf{P}_{\text{org}} - \mu|)} \cdot 2^{\mathbf{Q}[i]}), \tag{9}$$

where the quantization step is $s = \frac{\max(|\mathbf{P}-\mu|)}{2}$.

It is noted that different points may have the same QL, and hence we perform one quantization on those points with the same QL to accelerate the point-wise quantization, as shown in ***Algorithm 1***. Specifically, $\mathbf{P}_{\text{org}}$ is first divided into multiple partial groups $\mathbf{P}^k_{\text{org}}$ according to the QL value. Subsequently, each $\mathbf{P}^k_{\text{org}}$ is quantized into $\bar{\mathbf{P}}^k_{\text{org}}$ and further inverse-quantized into a temporary reconstructed $\hat{\mathbf{P}}^k_{\text{rec}}$. Finally, all temporary groups $\hat{\mathbf{P}}^k_{\text{rec}}$ are merged into one $\hat{\mathbf{P}}_{\text{rec}}$ and quantized using the maximum QL to obtain $\bar{\mathbf{P}}_{\text{qua}}$. In the experiments, the *float32*-type $\mathbf{P}^k_{\text{org}}$ is converted into a *double*-type to avoid the precision errors resulting from the normalization operation. Figure 4 provides a toy example of our quantization algorithm.

3.3 Proposed *lpcJRCD* Framework

Given an LPC $\mathbf{P}_{\text{org}} \in \mathbf{R}^{N\times 4}$, our *lpcJRCD* model aims to predict the maximum point-wise QL that has ignorable impacts on perception accuracy. To this end, we use the proposed dataset to construct a learning-based model $\mathcal{F}_{\text{lpcJRCD}}$ to predict the optimal point-wise $\mathbf{Q}^*$, where $\mathbf{Q}^*[i] \in \mathbf{L} = \{8, 9, \cdots, 23\}$. To accurately predict $\mathbf{Q}^*$, sparse and point convolution are employed to extract descriptive features.

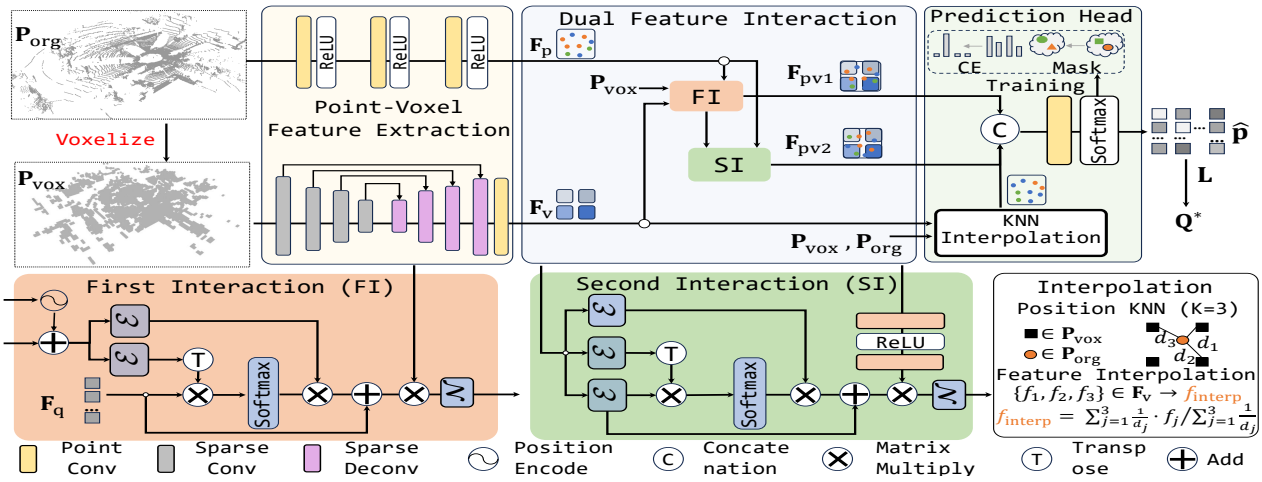

Figure 5: Pipeline of the proposed point-wise *lpcJRCD* prediction module.

Moreover, we design a dual feature interaction (DFI) module to effectively fuse latent features. As a result, our $\mathcal{F}_{\text{lpcJRCD}}$ is formulated as:

$$\mathbf{Q}^* = \mathcal{F}_{\text{lpcJRCD}}(\mathbf{P}_{\text{org}}) = \mathcal{F}_{\text{pred}}(\mathcal{F}_{\text{dfi}}([\mathcal{F}_{\text{sparse}}(\cdot), \mathcal{F}_{\text{point}}(\cdot)])), \quad (10)$$

where $\mathcal{F}_{\text{sparse}}$ denotes a sparse convolution module, $\mathcal{F}_{\text{point}}(\cdot)$ denotes a point convolution module, $\mathcal{F}_{\text{dfi}}(\cdot)$ denotes the proposed DFI module, and $\mathcal{F}_{\text{pred}}(\cdot)$ is the prediction head, The overall pipeline of $\mathcal{F}_{\text{lpcJRCD}}$ is also shown in Figure 5.

Point-Voxel Feature Extraction. Voxelization can transform unordered point clouds into ordered and structured data, facilitating subsequent hierarchical and global feature extraction. Therefore, we convert $\mathbf{P}_{\text{org}}$ into the voxel $\mathbf{P}_{\text{vox}}$, and use the sparse convolution to extract voxel feature:

$$\mathbf{F}_{\text{v}} = \mathcal{F}_{\text{sparse}}(\mathbf{P}_{\text{org}}) = \mathcal{F}_{\text{sdeconv}\times 4}(\mathcal{F}_{\text{sconv}\times 4}(\mathbf{P}_{\text{org}})), \quad (11)$$

where $\mathcal{F}_{\text{sconv}\times 4}$ denotes a 4-layer of sparse convolution, and $\mathcal{F}_{\text{sdeconv}\times 4}$ denotes a 4-layer of deconvolution with the skip connection.

Point convolution is able to preserve fine-grained local details, and hence we employ it to extract point features:

$$\mathbf{F}_{\text{p}} = \mathcal{F}_{\text{point}}(\mathbf{P}_{\text{org}}) = \mathcal{F}_{\text{pconv}}(\mathcal{F}_{\text{pconv}}(\mathcal{F}_{\text{pconv}}(\mathbf{P}_{\text{org}}))), \quad (12)$$

where $\mathcal{F}_{\text{pconv}}$ denotes a point convolution.

Dual Feature Interaction. To leverage both voxel and point features in Eq. (11) and Eq. (12), we devise the DFI module to fuse them into the dual point-voxel features $[\mathbf{F}_{\text{pv1}}, \mathbf{F}_{\text{pv2}}] = \mathcal{F}_{\text{dfi}}(\mathbf{F}_{\text{p}}, \mathbf{F}_{\text{v}})$. The first point-voxel interaction feature, $\mathbf{F}_{\text{pv1}}$, is extracted by cross-attention (CA):

$$\mathbf{F}_{\text{pv1}} = \mathcal{N}(\mathbf{F}_{\text{p}} \cdot (\texttt{Softmax}(\mathcal{E}(\mathbf{F}_{\text{q}}) \cdot \mathcal{E}(\mathbf{F}_{\text{v}} + \mathcal{F}_{\text{pos}}(\mathbf{P}_{\text{vox}}))^T) \cdot \mathcal{E}(\mathbf{F}_{\text{v}} + \mathcal{F}_{\text{pos}}(\mathbf{P}_{\text{vox}})) + \mathbf{F}_{\text{q}})^T), \in \mathbf{R}^{N\times|\mathbf{L}|}, \quad (13)$$

where $\mathbf{F}_{\text{q}} \in \mathbf{R}^{|\mathbf{L}|\times C}$ denotes a learnable query, $\mathcal{F}_{\text{pos}}$ denotes a `sin-cos` position encoding, $\mathcal{E}(\cdot)$ represents the embedding operation, and $\mathcal{N}(\cdot)$ represents the layer normalization operation. The second point-voxel interaction feature, $\mathbf{F}_{\text{pv2}}$, is extracted by self-attention (SA):

$$\mathbf{F}_{\text{pv2}} = \mathcal{N}(\mathcal{F}_{\text{pconv}}(\mathbf{F}_{\text{p}}) \cdot (\texttt{Softmax}(\mathcal{E}(\mathbf{F}'_{\text{v}}) \cdot \mathcal{E}(\mathbf{F}'_{\text{v}})^T) \cdot \mathcal{E}(\mathbf{F}'_{\text{v}}) + \mathbf{F}'_{\text{v}})^T, \in \mathbf{R}^{N\times|\mathbf{L}|}, \quad (14)$$

where $\mathbf{F}'_{\text{v}} = (\texttt{Softmax}(\mathcal{E}(\mathbf{F}_{\text{q}}) \cdot \mathcal{E}(\mathbf{F}_{\text{v}} + \mathcal{F}_{\text{pos}}(\mathbf{P}_{\text{vox}}))^T) \cdot \mathcal{E}(\mathbf{F}_{\text{v}} + \mathcal{F}_{\text{pos}}(\mathbf{P}_{\text{vox}})) + \mathbf{F}_{\text{q}})$.

Prediction Head. We finally use a light-weight prediction head $\mathcal{F}_{\text{pred}}$ to connect interpolated dual features, and predict the probability distribution of *lpcJRCD* by

$$\begin{aligned}\{\hat{\mathbf{p}}_i\}_{i=1}^N &= \mathcal{F}_{\text{pred}}(\mathbf{F}_{\text{pv1}}, \mathbf{F}_{\text{pv2}}, \mathbf{F}_{\text{v}}, \mathbf{P}_{\text{vox}}, \mathbf{P}_{\text{org}}) \\ &= \texttt{Softmax}(\mathcal{F}_{\text{pconv}}([\mathbf{F}_{\text{pv1}}, \mathbf{F}_{\text{pv2}}, \mathcal{F}_{\text{knn}}(\mathbf{F}_{\text{v}}, \mathbf{P}_{\text{vox}}, \mathbf{P}_{\text{org}})),\end{aligned} \quad (15)$$

where $\mathcal{F}_{\text{knn}}$ denotes the k-nearest neighbor (KNN) module to interpolate voxel features by weighting the $k = 3$ nearest points of $\mathbf{P}_{\text{org}}$ in $\mathbf{P}_{\text{vox}}$. The QL corresponding to the maximum probability value is the final prediction result:

$$\mathbf{Q}^*[i] = \mathbf{L}[\arg\max_k(\hat{\mathbf{p}}_i[k])]. \quad (16)$$

Mask-based Loss Function. To optimize the prediction accuracy of point-wise $\mathbf{Q}^*$ and the matching degree of $\bar{\mathbf{P}}^k_{\text{rec}}$, we propose to minimize the loss function, consisting of a *cross-entropy* (CE) loss for QL classification and a *Jaccard* loss for mask matching:

$$\mathbf{min} - \frac{\lambda_1}{N}\sum_{i=1}^{N}\log\hat{\mathbf{p}}_i[\mathcal{I}(\mathbf{Q}^{\text{lpcJRCD}}[i], \mathbf{L})] + \frac{\lambda_2}{|\mathbf{L}|}\sum_{k=1}^{|\mathbf{L}|}(1 - J(\mathbf{m}_k, \hat{\mathbf{m}}_k)), \quad (17)$$

where $\mathbf{Q}^{\text{lpcJRCD}}[i]$ denotes the ground-truth label, and $\mathcal{I}(\mathbf{Q}^{\text{lpcJRCD}}[i], \mathbf{L})$ denotes the index of $\mathbf{Q}^{\text{lpcJRCD}}[i]$ in $\mathbf{L}$. λ_1 and λ_2 represent two constant hyper-parameters. $\mathbf{m}_k$ and $\hat{\mathbf{m}}_k$ are the ground-truth and predicted masks, which are binary vectors with the same length as the number of points in $\mathbf{P}_{\text{org}}$, and they are used to indicate which points have QLs belonging to $\mathbf{L}[k]$. The i-th element of $\mathbf{m}_k$ is formulated as:

$$\mathbf{m}_k[i] = \begin{cases} 1, & \text{if } \mathbf{Q}^{\text{lpcJRCD}}[i] == \mathbf{L}[k] \\ 0, & \text{else} \end{cases}. \quad (18)$$

4 Experimental Validations

In this section, we have conducted extensive experiments to validate the effectiveness of our point-wise quantization method, our *lpcJRCD* prediction model, and our *lpcJRCD*-guided LPCC method.

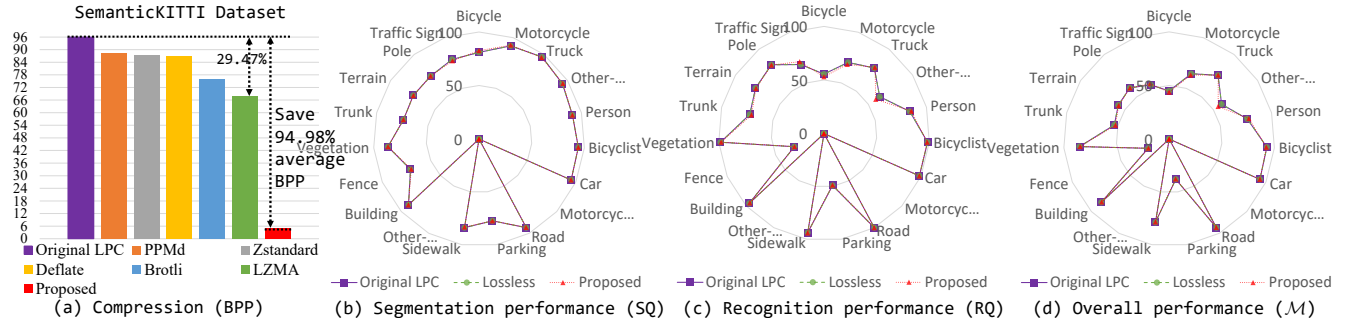

Figure 6: MVS-based compression results on the *SemanticKITTI* dataset. The segmentation and recognition results for different classes show that our method achieves similar performance to lossless method, while saving 94.98% in average BPP.

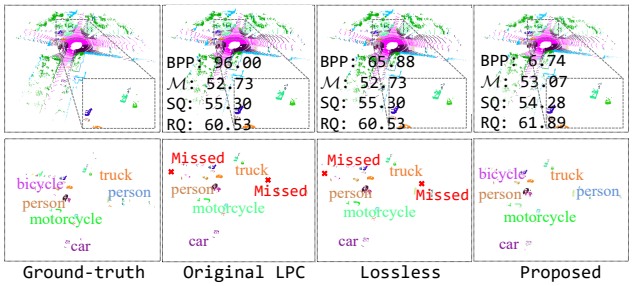

Figure 7: Visualization of segmentation and recognition results on the *SemanticKITTI* dataset.

Methods	$\mathcal{M}\uparrow$	RQ $\uparrow$	SQ $\uparrow$
Baseline	56.97	66.06	76.11
Baseline + $\mathbf{F}_{\text{pv1}}$	58.40	67.69	76.12
Baseline + $\mathbf{F}_{\text{pv1}}$ + $\mathbf{F}_{\text{pv2}}$	**59.41**	**68.74**	**76.24**

Table 1: Ablation experiment of our *lpcJRCD* model.

	Method	*nuScenes*				*SemanticKITTI*			
		BPP	M	RQ	SQ	BPP	M	RQ	SQ
HVS-	Original LPC	96.00	67.63	77.81	85.21	96.00	59.87	69.18	76.32
	PPMd (Lossless)	79.21	67.63	77.81	85.21	88.17	59.87	69.18	76.32
	Zstandard (Lossless)	79.03	67.63	77.81	85.21	87.46	59.87	69.18	76.32
	Deflate (Lossless)	77.91	67.63	77.81	85.21	86.76	59.87	69.18	76.32
	Brotli (Lossless)	79.09	67.63	77.81	85.21	75.74	59.87	69.18	76.32
	LZMA (Lossless)	67.00	67.63	77.81	85.21	67.71	59.87	69.18	76.32
	G-PCC	29.17	67.39	77.66	85.06	30.46	59.77	69.06	76.31
	OctAttention	27.20	67.39	77.66	85.06	28.40	59.77	69.06	76.31
	NERI	28.85	61.27	72.17	84.93	27.93	54.83	64.09	76.26
MVS-	*G-PCC**	3.99	67.38	77.66	85.09	4.95	59.41	68.74	76.24
	*OctAttention**	3.84	67.38	77.66	85.09	4.82	59.41	68.74	76.24
	*NERI**	9.87	60.72	71.92	84.83	9.33	54.39	63.71	76.14

Table 2: Compression comparisons of HVS- and MVS-based LPCC methods. '*' denotes *lpcJRCD*-based method.

4.1 Experiment Protocols

Experiment Settings. The dataset construction experiments are conducted on a computer with the CPU of *Intel Xeon W-226*. The panoptic segmentation and the *lpcJRCD* are implemented by *PyTorch* on a computing platform with the *NVIDIA A100* GPU and *Intel Xeon Gold 6226R*. For the training of our *lpcJRCD* model, λ_1 and λ_2 are set to 2 and 1, respectively. Our *lpcJRCD* model is trained using *AdamW* [Loshchilov and Hutter 2017] for 20 epochs with an initial learning rate of $10e{-}4$. The feature dimension C is set to 512. We use *Open3D* [Zhou, Jaesik, and Vladlen 2018] for LPC visualization.

Performance Measurements. Our proposed *lpcJRCD*-guided LPCC is evaluated by encoding efficiency and perception accuracy of segmentation and detection. We use the bit-per-point $\text{BPP} = \text{bits}/N$ to measure the compression efficiency, where bits represents the encoding bits of the input $\mathbf{P}_{\text{org}}$, and N represents the number of points in $\mathbf{P}_{\text{org}}$.

Perception accuracy is measured by $\mathcal{M}$ in Eq. (3), consisting of segmentation quality (SQ), and recognition quality (RQ). A smaller BPP indicates better compression performance, while a larger $\mathcal{M}$ signifies improved accuracy. The compression distortion caused by point-wise quantization is measured by the Chamfer distance (CD) [Wu et al. 2024].

Comparison Methods. We apply our proposed *lpcJRCD* model to octree-based methods, including *G-PCC* [Li, Gao, and Gao 2024], deep-learning-based *OctAttention* [Fu et al. 2022], and implicit neural representation (INR)-based NERI [Xue et al. 2024]. We compare them with five lossless compression methods, including *PPMd* [Saunders, Grant, and Müller 2018], *Zstandard* [Yann Collet. Accessed 2025], *Deflate* [Google. Accessed 2025], *Brotli* [Google. Accessed 2025], and *LZMA* [Pavlov. Accessed 2025].

4.2 Ablation Study

To verify the effectiveness of point-voxel features $\mathbf{F}_{\text{pv1}}$ and $\mathbf{F}_{\text{pv2}}$, we have conducted ablation experiments. As shown in Table 1, 'Baseline' directly predicts QL using features extracted from the sparse convolution. As seen, the dual point-voxel from our DFI module improves the prediction performance, resulting in better perception accuracy.

4.3 MVS-based Compression Performance

Segmentation and Recognition Applications. Table 2 provides the segmentation results of lossless and *lpcJRCD*-guided LPCCs. As shown, while lossless LPCCs maintain segmentation accuracy, they have the disadvantage of high BPPs. In contrast, *lpcJRCD*-guided LPCCs significantly reduces the BPP results while slightly improving performance

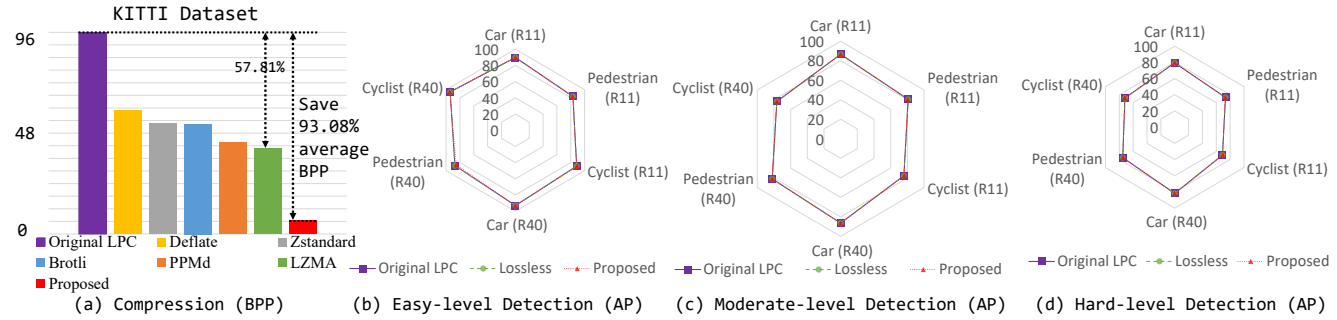

Figure 8: MVS-based compression results on the *KITTI* dataset. R40 and R11 denote average precision (AP) under 40 and 11 recall thresholds, respectively. Our method obtains similar detection accuracy, while achieving a 93.08% BPP reduction.

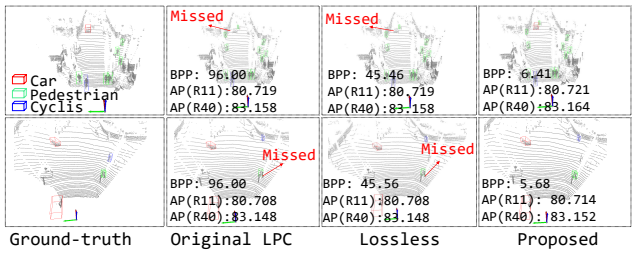

Figure 9: Visualization of object detection results on the *KITTI* dataset.

Methods	Car (R11)			Pedestrian (R11)			Cyclist (R11)		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
Lossless (40.50 BPP)	89.79	86.55	79.30	83.27	81.01	73.34	88.56	75.78	68.88
Proposed (6.64 BPP)	89.79	86.63	79.33	82.77	80.45	72.97	89.17	76.15	69.22
Methods	Car (R40)			Pedestrian (R40)			Cyclist (R40)		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
Lossless (40.50 BPP)	93.03	86.31	81.84	87.35	82.06	75.21	94.29	76.51	71.86
Proposed (6.64 BPP)	93.09	86.30	81.88	85.44	81.59	74.82	94.76	75.24	72.32

Table 3: Cross-dataset comparison on object detection between the lossless and *lpcJRCD*-guided methods. 'Easy', 'Mod.' (moderate), and 'Hard' indicate the difficulty levels.

of segmentation (see the reasons provided below Eq. (6)).

Figure 6 presents the average compression results of our proposed MVS-based LPCC (*i.e.*, '*Proposed*' denotes *lpcJRCD*-guided *OctAttention* in the remaining sections) in comparison with the lossless method (*i.e.*, '*Lossless*' denotes *LZMA*) for segmentation and recognition applications. As seen, our framework achieves significant compression improvements with negligible accuracy differences. Specifically, our method reduces the average BPP by 94.98%.

To qualitatively validate the scene understanding performance of our proposed *lpcJRCD*-guided LPCC, we present the segmentation and recognition results on the *SemanticKITTI* dataset in Figure 7. Different classes are marked with different colors, while instances are further distinguished by predicted class labels. As shown, the segmentation results of *lpcJRCD*-guided LPCC are nearly identical to those of the lossless method. Meanwhile, our *lpcJRCD*-guided method reduces the average BPP by approximately 89.77% compared to lossless compression. This reduction is mainly due to our method quantizing OOIs with lower QLs, which results in smaller BPPs.

Detection Application. To evaluate the generalization ability of our method, we have conducted cross-dataset experiments on the *KITTI* dataset and with one representative object detection model [Wu et al. 2022]. Table 3 presents the compression performance for the detection application. As shown, under similar detection accuracy in terms of average precision (AP), the BPP results of the lossless method is 40.50, while that of our proposed method is only 6.64.

Figure 8 shows the average compression results for object detection under the original, lossless, and *lpcJRCD*-guided LPCC. It can be observed that the proposed method significantly reduces the average bit-rate by 93.08%, achieving much greater bandwidth savings than lossless LPCCs while maintaining similar detection accuracy across different sample classes and difficulty levels.

Figure 9 presents qualitative performance of object detection results. As seen, our *lpcJRCD*-guided method reduces the BPP by approximately 85.90% (top) and 87.53% (bottom), respectively. Additionally, our method successfully detects objects that may be missed by the lossless compression. This further demonstrates the effectiveness of our method for 3D scene understanding in autonomous vehicles.

5 Conclusion

This paper presents an *lpcJRCD* model from the perspective of machine vision system (MVS), which significantly improves compression efficiency without compromising perception accuracy. Specifically, we have established one of the earliest MVS-based LPCC datasets, containing various objects with high significance in autonomous driving. In addition, we have constructed a dual feature interaction model to fuse point and voxel features. Based on the point-wise quality level prediction, we have further developed a new MVS-based LPCC framework with our proposed point-wise quantization to avoid *overhead bits explosion*. Experimental results on cross-dataset validate the effectiveness of our *lpcJRCD*-guided LPCC in improving compression efficiency for MVS-based applications.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62472290 and 62372306, and in part by the Natural Science Foundation of Guangdong Province under Grants 2024A1515011972, and 2023A1515011197.

References

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.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuScenes: A multimodal dataset for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition*, 11621–11631.

Chen, S.; Yang, B.; Xia, Y.; Cheng, M.; Shen, S.; and Wang, C. 2024. Bridging LiDAR Gaps: A Multi-LiDARs Domain Adaptation Dataset for 3D Semantic Segmentation. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Chen, Z.; Qian, Z.; Wang, S.; and Chen, Q. 2022. Point Cloud Compression with Sibling Context and Surface Priors. In *Springer European Conference on Computer Vision (ECCV)*, 744–759.

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.

Fan, T.; Gao, L.; Xu, Y.; Wang, D.; and Li, Z. 2024. Multiscale latent-guided entropy model for lidar point cloud compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(12): 7857–7869.

Fu, C.; Li, G.; Song, R.; Gao, W.; and Liu, S. 2022. OctAttention: Octree-Based Large-Scale Contexts Model for Point Cloud Compression. *AAAI Conference on Artificial Intelligence (AAAI)*, 36(1): 625–633.

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. Accessed 2025. Deflate. https://github.com/ebiggers/libdeflate,.

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

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

Jin, J.; Zhang, X.; Fu, X.; Zhang, H.; Lin, W.; Lou, J.; and Zhao, Y. 2021. Just noticeable difference for deep machine vision. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(6): 3452–3461.

Li, G.; Gao, W.; and Gao, W. 2024. MPEG geometry-based point cloud compression (G-PCC) standard. In *Springer Point Cloud Compression: Technologies and Standardization*, 135–165.

Liu, L.; Hu, Z.; and Zhang, J. 2023. PCHM-Net: A new point cloud compression framework for both human vision and machine vision. In *IEEE International Conference on Multimedia and Expo (ICME)*, 1997–2002.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 1–11.

Ma, G.; Wei, H.; Lin, R.; and Wu, J. 2024. A new guaranteed outlier removal method based on plane constraints for large-scale LiDAR point cloud registration. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 6868–6876.

Marcuzzi, R.; Nunes, L.; Wiesmann, L.; Behley, J.; and Stachniss, C. 2023. Mask-based panoptic lidar segmentation for autonomous driving. *IEEE Robotics and Automation Letters*, 8(2): 1141–1148.

Milioto, A.; Behley, J.; McCool, C.; and Stachniss, C. 2020. Lidar panoptic segmentation for autonomous driving. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8505–8512.

Pavlov., I. Accessed 2025. LZMA: Lempel–Ziv–Markov Chain Algorithm. http://7-zip.org/sdk.html.

Que, Z.; Lu, G.; and Xu, D. 2021. VoxelContext-Net: An Octree based Framework for Point Cloud Compression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6042–6051.

Saunders, W. R.; Grant, J.; and Müller, E. H. 2018. A domain specific language for performance portable molecular dynamics algorithms. *Elsevier Computer Physics Communications*, 224: 119–135.

Song, R.; Fu, C.; Liu, S.; and Li, G. 2023. Efficient Hierarchical Entropy Model for Learned Point Cloud Compression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 14368–14377.

Sun, L.; Wang, J.; Shi, Y.; Zhu, Q.; Yin, B.; and Ling, N. 2023a. Octree-Based Temporal-Spatial Context Entropy Model for LiDAR Point Cloud Compression. In *IEEE International Conference on Visual Communications and Image Processing (VCIP)*, 1–5.

Sun, X.; Wang, M.; Du, J.; Sun, Y.; Cheng, S. S.; and Xie, W. 2023b. A Task-Driven Scene-Aware LiDAR Point Cloud Coding Framework for Autonomous Vehicles. *IEEE Transactions on Industrial Informatics*, 19(8): 8731–8742.

Teeti, I.; Khan, S.; Shahbaz, A.; Bradley, A.; Cuzzolin, F.; and De Raedt, L. 2022. Vision-based Intention and Trajectory Prediction in Autonomous Vehicles: A Survey. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 5630–5637.

Google. Accessed 2025. Brotli. https://github.com/google/brotli.

Yann Collet. Accessed 2025. Zstandard. http://github.com/facebook/zstd.

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, M.; Huang, R.; Dong, H.; Lin, D.; Song, Y.; and Xie, W. 2024. msLPCC: A Multimodal-Driven Scalable Framework for Deep LiDAR Point Cloud Compression. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, 5526–5534.

Wang, M.; Huang, R.; Liu, Y.; Li, Y.; and Xie, W. 2025a. suLPCC: A novel LiDAR point cloud compression framework for scene understanding tasks. *IEEE Transactions on Industrial Informatics*, 21(5): 3816–3827.

Wang, M.; Huang, R.; Xie, W.; Ma, Z.; and Ma, S. 2025b. Compression Approaches for LiDAR Point Clouds and Beyond: A Survey. *ACM Transactions on Multimedia Computing, Communications and Applications*, 1–30.

Wang, M.; Ngan, K. N.; and Li, H. 2016. Low-delay rate control for consistent quality using distortion-based Lagrange multiplier. *IEEE Transactions on Image Processing*, 25(7): 2943–2955.

Wiesmann, L.; Milioto, A.; Chen, X.; Stachniss, C.; and Behley, J. 2021. Deep compression for dense point cloud maps. *IEEE Robotics and Automation Letters*, 6(2): 2060–2067.

Wu, H.; Deng, J.; Wen, C.; Li, X.; Wang, C.; and Li, J. 2022. CasA: A cascade attention network for 3-D object detection from LiDAR point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–11.

Wu, H.; Li, Q.; Wen, C.; Li, X.; Fan, X.; and Wang, C. 2021. Tracklet Proposal Network for Multi-Object Tracking on Point Clouds. In *nternational Joint Conference on Artificial Intelligence (IJCAI)*, 1165–1171.

Wu, Y.; Zhao, M.; Li, K.; Quan, W.; Yu, T.; Yang, J.; Jia, X.; and Yan, D.-M. 2024. CMG-Net: Robust Normal Estimation for Point Clouds via Chamfer Normal Distance and Multi-Scale Geometry. volume 38, 6171–6179.

Xie, L.; Gao, W.; Zheng, H.; and Li, G. 2024a. Roi-guided point cloud geometry compression towards human and machine vision. In *ACM International Conference on Multimedia (ACM MM)*, 3741–3750.

Xie, L.; Gao, W.; Zheng, H.; and Li, G. 2024b. SPCGC: scalable point cloud geometry compression for machine vision. In *IEEE International Conference on Robotics and Automation (ICRA)*, 17272–17278. IEEE.

Xue, R.; Li, J.; Chen, T.; Ding, D.; Cao, X.; and Ma, Z. 2024. NeRI: Implicit Neural Representation of LiDAR Point Cloud Using Range Image Sequence. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8020–8024.

You, K.; and Gao, P. 2021. Patch-based deep autoencoder for point cloud geometry compression. In *ACM Multimedia Asia (MMAsia)*, 1–7.

Yu, Y.; Zhang, W.; Yang, F.; and Li, G. 2023. Rate-distortion optimized geometry compression for spinning LiDAR point cloud. *IEEE Transactions on Multimedia*, 25: 2993–3005.

Zhang, Q.; Wang, S.; Zhang, X.; Ma, S.; and Gao, W. 2021. Just recognizable distortion for machine vision oriented image and video coding. *Springer International Journal of Computer Vision*, 129(10): 2889–2906.

Zhang, Y.; Lin, H.; Sun, J.; Zhu, L.; and Kwong, S. 2023. Learning to Predict Object-Wise Just Recognizable Distortion for Image and Video Compression. *IEEE Transactions on Multimedia*, 26: 5925–5938.

Zhou, Q.-Y.; Jaesik, P.; and Vladlen, K. 2018. Open3D: A Modern Library for 3D Data Processing. *arXiv:1801.09847*.

Zhou, X.; Qi, C. R.; Zhou, Y.; and Anguelov, D. 2022. RIDDLE: LiDAR Data Compression with Range Image Deep Delta Encoding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 17212–17221.