BitRIC: Efficient Neural Compression of LiDAR Range Images via Hierarchical Bitplanes

Kang You[1], Tong Chen[1], Dandan Ding[2], M. Salman Asif[3], and Zhan Ma[1(✉)]

[1] School of Electronic Science and Engineering, Nanjing University, Nanjing, China
youkang@smail.nju.edu.cn, {chentong,mazhan}@nju.edu.cn
[2] School of Information Science and Technology, Hangzhou Normal University, Hangzhou, China
[3] Department of Electrical and Computer Engineering, UC Riverside, Riverside, USA

Abstract. The rapid evolution of 3D scene understanding relies heavily on high-resolution LiDAR, which generates massive data volumes that impose significant bottlenecks on transmission and storage. While Range Image Compression (RIC) provides a more structured and computationally efficient alternative to conventional Point Cloud Compression (PCC), existing RIC frameworks frequently fail to achieve an optimal trade-off between coding efficiency and real-time performance. To bridge this gap, we propose BitRIC, a novel learning-based framework tailored for high-performance LiDAR range image compression. BitRIC decomposes raw range images into hierarchical bitplanes and processes them through a coarse-to-fine autoregressive entropy model. At its core, a shared contextual backbone leverages cross-bitplane dependencies to inform two specialized pathways: a Probability Estimation Module (PEM) that predicts probability maps for lossless entropy coding, and a One-Step Refinement Module (ORM) that reconstructs residual details during bitplane truncation for lossy compression. Extensive experiments on the Waymo Open Dataset and SemanticKITTI demonstrate that BitRIC achieves superior rate-distortion performance, yielding over 60% BD-BR savings compared to standard image codecs, while offering higher downstream task accuracy. Furthermore, BitRIC supports real-time lossy and near-real-time lossless processing, making it a highly practical candidate for real-world applications.

Keywords: LiDAR · Range image · Data compression

1 Introduction

LiDAR (Light Detection and Ranging) technology has emerged as a cornerstone of 3D scene understanding, fueling the rapid evolution of autonomous driving [20, 41], mobile robotics [9, 18, 38], and smart city infrastructure [26]. By providing precise spatial measurements, LiDAR enables robust perception under diverse environmental and lighting conditions, performing a feat that is often challenging for passive camera-based systems [15]. However, the industry's

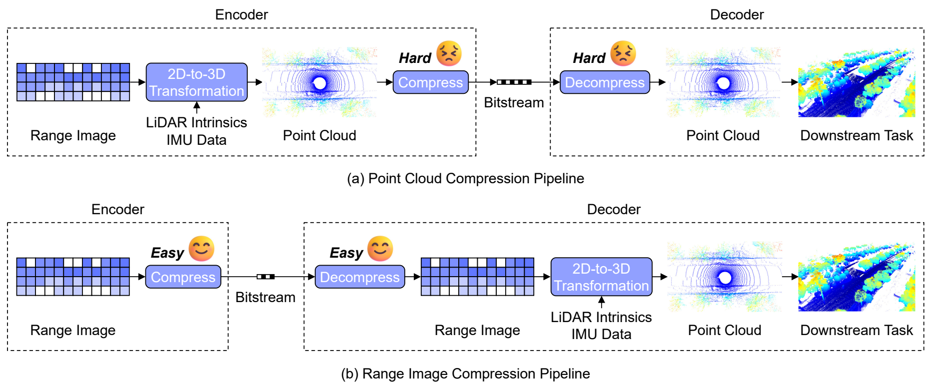

Fig. 1: Point cloud vs. range image compression. (a) Point Cloud Compression (PCC) requires 2D-to-3D transformation, resulting in sparse data that lacks topological connectivity, which limits compression performance. (b) Range Image Compression (RIC) leverages a regular, grid-like representation, making it easier to achieve superior coding efficiency and lightweight deployment. Although RIC requires auxiliary metadata (*e.g.*, LiDAR intrinsics and IMU data), its impact on the total data volume is marginal.

shift toward high-resolution configurations has triggered an unprecedented explosion in data volume. For mission-critical applications like Vehicle-to-Everything (V2X) communication [26], this "data deluge" creates a severe bottleneck in transmission bandwidth and onboard storage capacity, desiring highly efficient and real-time compression solutions [1].

Modern LiDAR sensors typically operate by emitting laser pulses across a fixed number of vertical channels and horizontal angular positions, employing either mechanical, hybrid solid state or all-solid-state scanning mechanisms [21]. This scanning geometry intrinsically structures the raw sensor data into a 2D grid indexed by azimuth and elevation angles of laser scanners, a representation commonly referred to as a range image [6, 14, 28]. In this format, each pixel encodes the radial distance (range) from the sensor origin to the intercepted surface, while auxiliary channels often incorporate features such as intensity [19].

Conventionally, these raw range measurements are projected into 3D Cartesian space to produce a point cloud (*a.k.a.*, 2D-to-3D transformation [37]), which is defined as an unordered set of 3D coordinates (x, y, z). As shown in Fig. 1(a), the Point Cloud Compression (PCC) pipeline requires 2D-to-3D transformation to produce 3D points [27], resulting in sparse data that lacks topological connectivity and poses significant challenges for data compression. Current point cloud compression methods often resort to multi-scale octrees or specialized sparse tensor representations, which impose non-trivial computational overhead and memory footprints. Paradoxically, a growing trend in high-performance compression involves mapping these sparse 3D points back into spherical [22, 25] or cylindrical [10] coordinates to regain structural regularity, mimicking the original range image format. Such a redundant "range-to-point-to-range" cycle not only complicates the compression pipeline but also fails to fully exploit the native structural regularity of the sensor's original output.

Conversely, as depicted in Fig. 1(b), Range Image Compression (RIC) leverages a compact representation with regular topology. Motivated by these observations, we argue that performing compression directly in the range image domain offers a more streamlined and computationally efficient alternative to conventional PCC.

Despite its potential, existing RIC frameworks still face several critical challenges. Traditional image codecs (*e.g.*, JPEG2000 [30]), while efficient for natural images, often struggle with the unique statistical properties of LiDAR data, such as high-frequency depth discontinuities at object boundaries and the non-uniform distribution of range values. Conversely, emerging learning-based RIC methods [16, 23, 37, 43] fail to strike an optimal balance between low latency and high performance, often necessitating a trade-off where reconstruction accuracy is sacrificed for real-time capability, or vice versa.

To bridge this gap, we propose BitRIC, a novel learning-based framework specifically engineered for high-performance LiDAR range image compression. BitRIC first decomposes the input range image into a sequence of bitplanes, and then a coarse-to-fine autoregressive entropy model is introduced to effectively exploit inter-plane redundancies. A shared contextual backbone extracts rich spatial priors to drive two specialized pathways: (i) a Probability Estimation Module (PEM) that predicts bit-wise distributions for lossless entropy coding, and (ii) a One-Step Refinement Module (ORM) that reconstructs residual details from truncated bitplanes to facilitate high-quality lossy compression. Extensive evaluations on the Waymo Open Dataset [29] and SemanticKITTI [3] demonstrate that BitRIC achieves superior rate-distortion performance, delivering over 60% BD-BR savings compared to traditional codecs while significantly enhancing downstream task accuracy. Notably, BitRIC supports real-time lossy and near-real-time lossless processing, offering a versatile and practical solution for real-world applications.

The main contributions of this paper are highlighted as follows:

- We propose BitRIC, a novel neural compression framework that leverages bitplane-based representation for LiDAR range images, achieving superior performance in both rate-distortion efficiency and downstream task accuracy.
- We design a unified dual-path architecture that seamlessly supports both lossless and lossy compression regimes, providing a versatile solution for diverse bandwidth and fidelity requirements within a single model.
- Extensive evaluations demonstrate that our method achieves real-time processing for lossy compression and near-real-time performance for lossless scenarios, offering a compelling candidate for real-world applications.

2 Related Work

Range Image Compression (RIC) generally falls into two paradigms: traditional rule-based approaches and emerging neural compression models, both of which are introduced below.

Rules-based Approaches. Early attempts primarily adapted general-purpose image codecs (*e.g.*, JPEG, PNG) to compress range image data [11, 34, 35]. In recent years, more representative methods tailored specifically for LiDAR characteristics have emerged. For instance, Yu *et al.* [7] proposed a LiDAR compression method leveraging iterative plane fitting and cross-frame plane reuse to achieve real-time efficiency. This framework was further extended in RCPCC [6], where a surface model is adopted to better capture the spatial structure of points in the range image. R-PCC [36] employs RANSAC [8] to extract ground points from the range image, followed by segmenting the remaining points into compact sub-clusters to facilitate efficient compression. While Zhao *et al.* [42] leverage learning-based semantic segmentation to establish a semantic prior representation comprising labels, predictions, and residuals, the core compression mechanism remains anchored in the conventional FPZIP [17] algorithm. Despite their efficiency, rules-based approaches often struggle to fully exploit complex data redundancies, resulting in sub-optimal rate-distortion performance.

Neural Compression Models. Neural compression methods have gained prominence by replacing traditional hand-crafted codecs with end-to-end trainable networks. Several studies [23, 33] employ Variational Autoencoders (VAEs) to compress range image data. Although these approaches achieve real-time efficiency, the VAE architecture often struggles to reconstruct fine-grained geometric details accurately, even when incorporating binary masks to identify invalid pixels. To enhance reconstruction fidelity, alternative methods [2, 32] utilize Recurrent Neural Networks (RNNs) to progressively compress data via residual prediction. Similarly, RIDDLE [43] introduces a deep delta encoding scheme that predicts pixel values in raster-scan order, leveraging contextual information from both spatial and temporal scans. While these autoregressive and recurrent schemes [2, 32, 43] demonstrate superior rate-distortion performance, their sequential dependency inherently limits parallelization, resulting in high computational latency. Recently, Implicit Neural Representations (INR) [16, 37] have emerged as a promising alternative. These methods represent each range image by overfitting a compact neural network, whose parameters are subsequently quantized and encoded into the bitstream. However, since the encoding phase requires a dedicated optimization process for each frame, it remains computationally expensive.

In summary, existing RIC methods struggle to strike an optimal balance between *coding efficiency* and *computational throughput*. This research gap motivates the need for a robust 2D compression framework capable of achieving high-performance compression without compromising real-time efficiency.

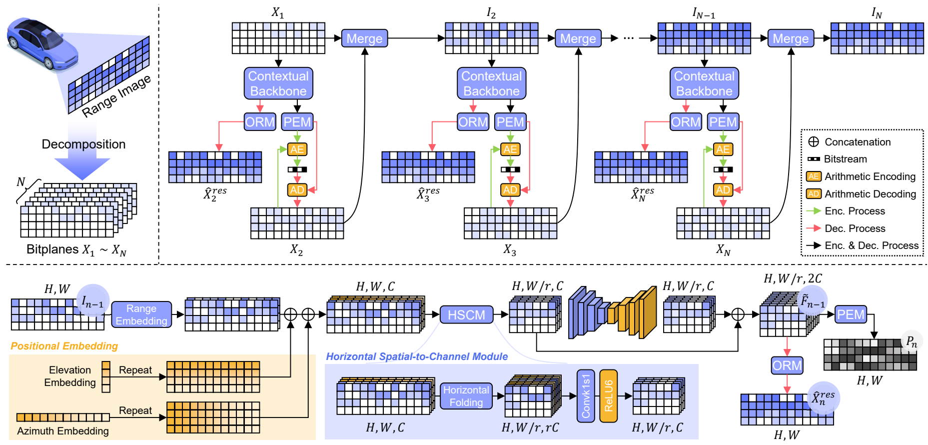

Fig. 2: BitRIC. Upper left: The input range image is initially decomposed into N hierarchical bitplanes; Upper right: An autoregressive encoding scheme is employed, where each bitplane is compressed by conditioning on the previously encoded bitplanes; Bottom: The contextual backbone structure is shown, where the output feature $\tilde{F}_{n-1}$ is either fed into the Probability Estimation Module (PEM) for lossless coding or the One-Step Refinement Module (ORM) to predict the residual details.

3 Methodology

The framework of the proposed BitRIC is shown in Fig. 2. Initially, the input range image is decomposed into N hierarchical bitplanes. We then develop a convolutional neural network to encode these bitplanes autoregressively, processing from the Most Significant Bit (MSB) plane (coarse structure) to the Least Significant Bit (LSB) plane (high-frequency details). In lossless mode, all N bitplanes are encoded via a Probability Estimation Module (PEM) and arithmetic coding to ensure full reconstruction. In lossy mode, the encoding is truncated at a predefined bitplane; a One-step Refinement Module (ORM) is then integrated into the decoding stage to predict the remaining planes in a single inference pass.

3.1 Hierarchical Bitplane Decomposition

Given a range image $I_N \in \mathbb{R}^{H \times W}$, we assume each pixel $I_N(i,j)$ is expressed as an N-bit non-negative integer[4], such that $I_N(i,j) \in \{0, 1, \ldots, 2^N - 1\}$. To facilitate efficient compression, we decompose I_N into an ordered sequence of N binary bitplanes, denoted as $\mathcal{X} = \{X_1, \ldots, X_N\}$. Mathematically, the binary state of a pixel at position (i,j) in the n-th bitplane is formulated as:

$$X_n(i,j) = \left\lfloor \frac{I_N(i,j)}{2^{N-n}} \right\rfloor \bmod 2, \quad n \in \{1, \ldots, N\}. \tag{1}$$

[4] In practice, range measurements from most commercial LiDAR sensors are natively encoded as unsigned integers (*e.g.*, representing distance in millimeters) [5].

In this formulation, X_1 represents the Most Significant Bit (MSB) plane, capturing the global geometric structure, while X_N denotes the Least Significant Bit (LSB) plane, which typically contains high-frequency details and sensor noise. The original range image can be losslessly recovered through weighted reconstruction, *i.e.*, $I_N(i,j) = \sum_{n=1}^{N} X_n(i,j) \cdot 2^{N-n}$. Note that invalid pixels (corresponding to invalid returns from laser measurements) are coded as zeros.

3.2 Bitplane-based Entropy Model

Given the bitplane decomposition $\mathcal{X}$, we propose a neural entropy model to efficiently compress these bitplanes. This section first introduces the lossless compression framework, as its autoregressive structure serves as the underpinning of our approach. We then extend this framework to a lossy mode.

Lossless Mode. To compress the bitplane sequence $\mathcal{X}$, we construct a deep parametric model to approximate the underlying probability distribution $p(\mathcal{X})$. Theoretically, minimizing the compression bitrate R is equivalent to minimizing the cross-entropy between the estimated distribution $p(\mathcal{X})$ and the actual distribution $q(\mathcal{X})$:

$$R = \mathbb{E}_{\mathcal{X} \sim q(\mathcal{X})}[-\log_2 p(\mathcal{X})]. \tag{2}$$

To capture inter-plane redundancies, the joint probability $p(\mathcal{X})$ is factorized into a coarse-to-fine autoregressive chain. This allows the model to progressively predict the distribution of the current bitplane X_n conditioned on all previously decoded bitplanes $X_{<n}$:

$$p(\mathcal{X}) = p(X_1, \ldots, X_N) = p(X_1) \prod_{n=2}^{N} p(X_n \mid X_{<n}) = p(X_1) \prod_{n=2}^{N} p(X_n \mid I_{n-1}), \tag{3}$$

where $I_{n-1} = \sum_{k=1}^{n-1} X_k \cdot 2^{N-k}$ denotes the partial reconstruction derived from $X_{<n}$. Since I_{n-1} is a bijective representation of $X_{<n}$, conditioning on I_{n-1} is mathematically equivalent to conditioning on the preceding bitplanes. As each element takes binary values (*i.e.*, $X_n(i,j) \in \{0,1\}$), a neural network is employed to output a probability map P_n for the n-th bitplane. Each entry $P_n(i,j)$ represents the Bernoulli parameter $\pi_{n,i,j} = p(X_n(i,j) = 1 \mid I_{n-1})$. Finally, the estimated distribution is integrated into an arithmetic coder for entropy coding.

Lossy Mode. Building upon the lossless backbone, the lossy mode introduces a bit-depth truncation mechanism. Instead of encoding all N bitplanes, we let the encoder only transmit the first M planes ($M < N$), where M is a hyperparameter controlling the trade-off between bitrate and reconstruction quality. The bits of the remaining bitplanes are discarded to save bandwidth. To compensate for the information loss, the truncated planes $X_{M+1:N}$ are jointly estimated in a single inference pass, conditioned on the received coarse-scale context $X_{1:M}$, thereby enhancing reconstruction fidelity.

3.3 Contextual Backbone

As illustrated in Fig. 2, the proposed neural network architecture centers on a contextual backbone designed to extract prior knowledge from the partial reconstruction I_{n-1}. This backbone serves as a shared feature extractor that informs two specialized pathways: i) Probability Estimation Module (PEM) that predicts the probability map P_n for the target bitplane X_n, enabling lossless entropy coding; ii) One-Step Refinement Module (ORM) that estimates the residual details $\hat{X}_n^{res}$ when subsequent bitplanes are truncated.

Range Embedding. As shown in Fig. 2, the contextual backbone first maps I_{n-1} into a latent feature space via an embedding layer:

$$F_{n-1}^{range} = \mathrm{Emb}(I_{n-1}), \tag{4}$$

where $\mathrm{Emb}(\cdot)$ denotes a learnable lookup table that transforms integer range values into continuous representations $F_{n-1}^{range} \in \mathbb{R}^{H\times W\times C^{range}}$.

Positional Embedding. Considering that the LiDAR data distribution is correlated with scanning angles (*e.g.*, lower beams hitting the proximal ground), we introduce learnable positional embeddings $F^{az} \in \mathbb{R}^{H\times 1\times C^{az}}$ and $F^{el} \in \mathbb{R}^{1\times W\times C^{el}}$. These parameters capture the sensor's scanning priors within the row and column indices. The final enhanced feature map F_{n-1} is constructed by concatenating the contextual features with the positional embeddings:

$$F_{n-1} = F_{n-1}^{range} \oplus \mathrm{Repeat}(F^{az}) \oplus \mathrm{Repeat}(F^{el}), \tag{5}$$

where $\oplus$ denotes concatenation; $\mathrm{Repeat}(\cdot)$ duplicates the positional embeddings along the spatial axes to match the resolution of F_{n-1}^{range}. For notational simplicity, we define the augmented feature as $F_{n-1} \in \mathbb{R}^{H\times W\times C}$, where $C = C^{range} + C^{az} + C^{el}$.

Horizontal Folding. Typically, LiDAR horizontal resolution significantly exceeds its vertical counterpart (*e.g.*, 2048 columns vs. 64 rows [29]). To reduce computational complexity, we introduce a Horizontal Spatial-to-Channel Module (HSCM) to downsample the horizontal axis. Specifically, the HSCM employs a spatial-to-channel folding operation with a downsampling ratio r, reorganizing $1\times r$ spatial neighborhoods into the channel dimension. Mathematically,

$$F_{n-1}^{fold}(i, j, k\cdot C + c) = F_{n-1}(i, j\cdot r + k, c),\ 0 \le c \le C-1,\ 0 \le k \le r-1, \tag{6}$$

where i and j denote the vertical and downsampled horizontal coordinates, respectively; c means the channel index; k indicates the horizontal offset. Finally, a 1×1 convolution paired with a ReLU6 activation aggregates the remapped feature $F_{n-1}^{fold} \in \mathbb{R}^{H\times \frac{W}{r}\times rC}$ to produce the refined output $\bar{F}_{n-1} \in \mathbb{R}^{H\times \frac{W}{r}\times C}$. We empirically set $r = 4$ to strike a balance between computational efficiency and the preservation of fine-grained horizontal structures.

U-Net-based Aggregation. To further extract spatial dependencies, a U-Net-like encoder-decoder architecture is applied to the downsampled feature map $\bar{F}_{n-1}$. This component facilitates the aggregation of global context while preserving local geometric structures through skip connections. The final shared

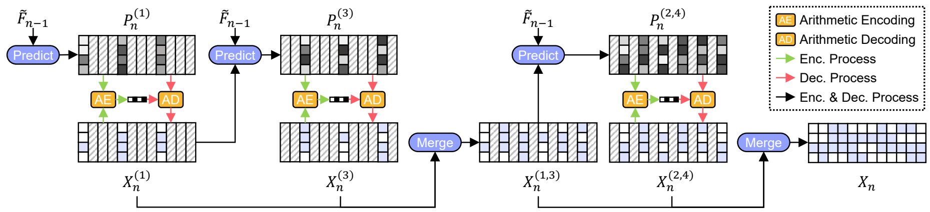

Fig. 3: Three-Stage Probability Estimation. The target bitplane X_n is partitioned into sub-groups $X_n^{(1)}$, $X_n^{(3)}$, and $X_n^{(2,4)}$, which are predicted in a cascaded manner.

prior feature $\tilde{F}_{n-1}$ is constructed by concatenating the original downsampled feature with the U-Net's output, formulated as:

$$\tilde{F}_{n-1} = \bar{F}_{n-1} \oplus \mathrm{UNet}(\bar{F}_{n-1}), \tag{7}$$

where $\tilde{F}_{n-1} \in \mathbb{R}^{H \times \frac{W}{r} \times 2C}$. This augmented feature serves as the common input for the subsequent PEM and ORM to guide probability estimation and residual refinement, respectively. Detailed specifications of the U-Net architecture are provided in the supplementary material.

3.4 Dual-Path Prediction

Probability Estimation Module (PEM). To balance the compression ratio and decoding efficiency, the PEM employs a three-stage scheme to predict the probability map P_n of the target bitplane X_n. Specifically, we partition X_n along the width dimension into three sub-groups: $X_n^{(1)}$, $X_n^{(3)}$, and $X_n^{(2,4)}$. As shown in Fig. 3, the prediction proceeds in three cascaded stages: (i) The probabilities for $X_n^{(1)}$ are directly estimated from $\tilde{F}_{n-1}$ via a lightweight prediction head; (ii) The ground-truth symbols of $X_n^{(1)}$ are embedded, processed via convolutions, and concatenated with $\tilde{F}_{n-1}$ to predict the distribution of $X_n^{(3)}$; (iii) The first two sub-groups are merged to form $X_n^{(1,3)}$ (odd columns), which is further embedded and fused with the base context $\tilde{F}_{n-1}$ to estimate the probabilities for the remaining odd columns $X_n^{(2,4)}$. By dividing the target bitplane into independent sets, the PEM effectively captures intra-plane spatial correlations while maintaining high efficiency with only three sequential steps.

One-Step Refinement Module (ORM). To handle bit-depth truncation, the ORM recovers the truncated bitplanes $X_{n:N}$ within a single inference step. We formulate the residual prediction as a continuous regression task, where the discarded bitplanes are aggregated and normalized to generate a ground-truth residual map $X_n^{res} \in [0,1]^{H \times W}$. Specifically, a sequence of convolutional layers processes the shared prior feature $\tilde{F}_{n-1}$ to estimate the continuous residual map $\hat{X}_n^{res}$. During decoding, the final reconstructed representation $\hat{I}_N$ is obtained by denormalizing the predicted residual back to its original integer scale. Please refer to the supplementary material for further details regarding this process and the network architecture.

3.5 Loss Function

BitRIC is optimized end-to-end by jointly minimizing bitrate R and distortion D. The bitrate R is defined as the cross-entropy across N bitplanes, *i.e.*, $R = \sum_{n=1}^{N} \mathbb{E}_{X_n \sim q(X_n)}[-\log_2 p(X_n)]$. The distortion is measured by the Mean Squared Error (MSE) between the ground-truth residual and its prediction, *i.e.*, $D = \sum_{n=1}^{N} \mathrm{MSE}(X_n^{res}, \hat{X}_n^{res})$. The total loss is formulated as $\mathcal{L} = R + D$. In particular, the conventional rate-distortion weighting parameter (*e.g.*, λ) is omitted, as the bitrate-quality trade-off is explicitly governed by the bit-depth truncation hyperparameter M rather than continuous loss scaling.

4 Experiment

4.1 Experimental Setting

Datasets. We evaluate the performance of BitRIC on two widely-used datasets: the Waymo Open Dataset (WOD) [29] and SemanticKITTI [3].

- WOD serves as our primary dataset as it is one of the few large-scale, widely recognized benchmarks that provides raw range data alongside downstream task labels. The dataset comprises 1,150 sequences, of which 798 are used for training and 202 for validation. Each 20-second sequence is captured at 10Hz using a 64-beam LiDAR, yielding range images with a resolution of 64×2,650. By leveraging the provided calibration metadata and ego-motion data, these range images are projected into accurate point clouds. In our experiments, we utilize only the first-return range images, cropped at a maximum range of 75m and quantized to 1mm precision.
- SemanticKITTI is employed for further evaluation, which is captured using a Velodyne HDL64E LiDAR and comprises 22 sequences totaling 43,504 frames. Following standard protocols, sequences 00∼10 are used for training, with sequence 08 being reserved for validation. Since SemanticKITTI provides point clouds rather than raw range images, we project the point clouds into spherical coordinates to generate pseudo range images with a resolution of 64×2,048. These pseudo-range images are cropped at a maximum range of 70m and quantized to 1mm precision.

Baselines. We assess BitRIC by comparing it with diverse baselines across two categories: (i) PCC methods, including rule-based (MPEG G-PCC [12]) and learning-based (RENO [40]) models; and (ii) RIC codecs, spanning traditional (JPEG2000 [30]) and specialized (RPCC [36], RCPCC [6]) approaches. G-PCC is implemented via the TMC13v23 reference software (octree mode). For JPEG2000, we modulate the bit-depth of the quantized range values to obtain multiple compression rates.

Evaluation Metrics. Since the primary utility of range images lies in their conversion to point clouds for downstream processing, the distortion is evaluated in 3D space, using three geometric metrics: point-to-point (D1) PSNR, point-to-plane (D2) PSNR, and Chamfer Distance (CD). The PSNR peak value is fixed

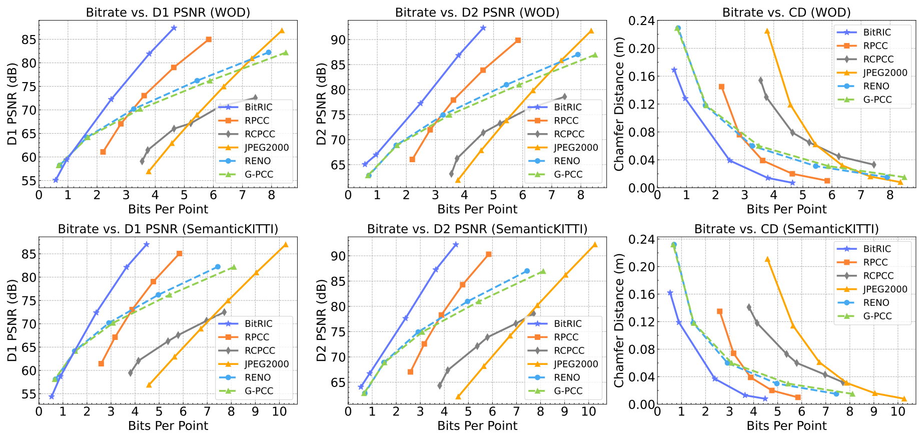

Fig. 4: Rate-distortion performance comparison on WOD (first row) and SemanticKITTI (second row) dataset.

at 59.70. The Bjøntegaard Delta Bitrate (BD-BR) [4] is used to assess rate-distortion performance, where the bitrate is measured in bits per point (Bpp). Furthermore, we evaluate downstream task performance (*e.g.*, 3D object detection) to investigate the impact of compression artifacts on practical applications.
Implementation Details. BitRIC is implemented using Python 3.10 and PyTorch 2.1. It is trained for 70,000 steps using the Adam optimizer [13] with an initial learning rate of $5{\times}10^{-4}$ and a batch size of 1. For the rate-distortion evaluation, BitRIC employs spatial downsampling (*e.g.*, by a factor of 2 or 4 in the horizontal axis) on the input range images at low bit rates (*i.e.*, $<$1 Bpp) to achieve enhanced trade-offs. For downstream task evaluation, we employ the CenterPoint [39] model within the OpenPCDet [31] framework. All experiments are executed on a workstation equipped with an Intel Xeon Platinum 8352V CPU and one NVIDIA GeForce RTX 4090 GPU.

4.2 Quantitative Comparison

Rate-Distortion Performance. Figure 4 and Tab. 1 illustrate the compression efficiency of the evaluated methods. Under both lossy and lossless configurations, BitRIC consistently achieves superior performance across all metrics. While PCC methods such as G-PCC and RENO show competitive results at low bitrates (*e.g.*, D1 PSNR $<$ 65 dB), this operating regime is often insufficient for high-fidelity downstream applications [40], rendering such low-bitrate gains less practical. Furthermore, the inherent sparsity of point cloud representations restricts these methods from scaling effectively to higher bitrates. In contrast, BitRIC maintains its advantage across the broader bitrate spectrum. Driven by an efficient bitplane-based autoregressive entropy model, our approach significantly outperforms existing RIC techniques. Specifically, BitRIC achieves substantial

Table 1: Compression performance comparison on WOD and SemanticKITTI datasets. For lossy compression, BD-BR (%) is reported for D1 and D2 metrics relative to the JPEG2000 anchor. For lossless compression, Bpp denotes bits per point, and Gain (%) represents the bitrate saving (or increase) relative to JPEG2000.

		WOD				SemanticKITTI			
		Lossy		Lossless		Lossy		Lossless	
Method	Input Type	D1 ↓	D2 ↓	Bpp ↓	Gain ↓	D1 ↓	D2 ↓	Bpp ↓	Gain ↓
G-PCC [12]	Point Cloud	-44.90%	-44.74%	21.48	+59.82%	-59.32%	-58.79%	20.84	+28.96%
RENO [40]	Point Cloud	-47.30%	-47.14%	19.42	+44.49%	-61.12%	-60.61%	18.73	+15.90%
JPEG2000 [30]	Range Image	-00.00%	-00.00%	13.44	-00.00%	-00.00%	-00.00%	16.16	-00.00%
RCPCC [6]	Range Image	-3.35%	-5.16%	N/A	N/A	-14.21%	-16.10%	N/A	N/A
RPCC [36]	Range Image	-39.79%	-39.78%	N/A	N/A	-47.65%	-47.75%	N/A	N/A
BitRIC	Range Image	**-61.73%**	**-61.52%**	**6.05**	**-54.99%**	**-69.92%**	**-70.82%**	**8.43**	**-47.83%**

Table 2: Computational complexity comparison. We report the average encoding (Enc.) and decoding (Dec.) time per frame in seconds. For lossy compression, the reported values are averaged across all tested rate-distortion (RD) points.

		G-PCC	RENO	JPEG2000	RCPCC	RPCC	BitRIC
Lossy	Enc.	0.749s	0.064s	0.046s	0.245s	0.201s	0.071s
	Dec.	0.405s	0.066s	0.041s	0.171s	0.049s	0.087s
Lossless	Enc.	0.869s	0.242s	0.132s	N/A	N/A	0.144s
	Dec.	0.422s	0.288s	0.084s	N/A	N/A	0.175s

BD-rate reductions compared to the JPEG2000 baseline, yielding improvements of over 60% and 50% in lossy and lossless scenarios on the WOD dataset, respectively (see Tab. 1), further underscoring its robust coding efficiency.

Computational Complexity. Table 2 reports the average per-frame encoding and decoding times. Note that runtime comparisons should be interpreted as indicative of computational complexity, given implementation variations across hardware (*e.g.*, CPU vs. GPU). Results show that BitRIC achieves real-time processing (>10 fps) for lossy compression, outperforming other range-image-based methods such as RCPCC and RPCC, which cannot maintain real-time efficiency. BitRIC also significantly outperforms JPEG2000 in compression performance. In the lossless setting, only JPEG2000 maintains real-time decoding. Exploring acceleration via model quantization [24] for real-time lossless compression is left for future work.

4.3 Qualitative Analysis

Figure 5 illustrates the reconstructed samples of various RIC methods under both low-bitrate and high-precision scenarios, utilizing two representative LiDAR scans from the WOD dataset. As demonstrated, our proposed BitRIC achieves significantly higher reconstruction fidelity compared to existing benchmarks. For instance, as shown in the first row, BitRIC effectively preserves the structural contours of the vehicle and pedestrian. In contrast, RPCC exhibits noticeable blurring around pedestrian geometries, while the reconstruction quality

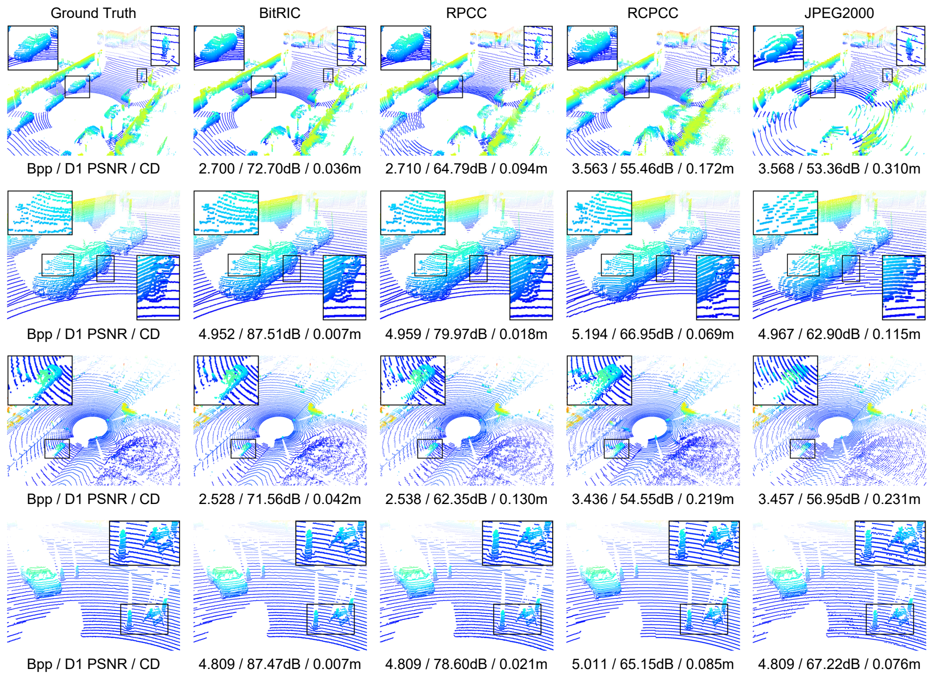

Fig. 5: Qualitative visualization. Range images are converted into point clouds to facilitate a more intuitive assessment. Rows 1 and 3 display compression results at lower bitrates, while rows 2 and 4 show results at higher bitrates. Point color denotes elevation relative to the ground, transitioning from blue (low) to red (high).

of RCPCC and JPEG2000 is severely degraded by compression artifacts. Under high-bitrate conditions, although BitRIC and RPCC appear qualitatively similar to the naked eye, quantitative analysis reveals a substantial performance gap. Specifically, BitRIC achieves a Chamfer Distance of only 7 mm, which is less than half that of RPCC, further validating its superior efficiency.

4.4 Downstream Task Evaluation

To investigate the practical implications of compression artifacts on autonomous driving applications, we evaluate 3D object detection performance using the CenterPoint [39] detector, which is pre-trained on uncompressed WOD range images. As illustrated in Fig. 6, BitRIC demonstrates exceptional robustness to compression noise. While traditional and specialized codecs (e.g., JPEG2000, RPCC, RCPCC) experience a precipitous drop in mean Average Precision (mAP) at lower bitrates, BitRIC maintains a performance highly comparable to the uncompressed raw baseline. For example, at $\sim$2 Bpp, BitRIC maintains a pedestrian (Level 1) detection mAP of 65.67 (vs. 67.77 for raw data), significantly exceeding RPCC (58.95), while RCPCC and JPEG2000 fail to yield meaningful detection results. The superior downstream performance of BitRIC can be

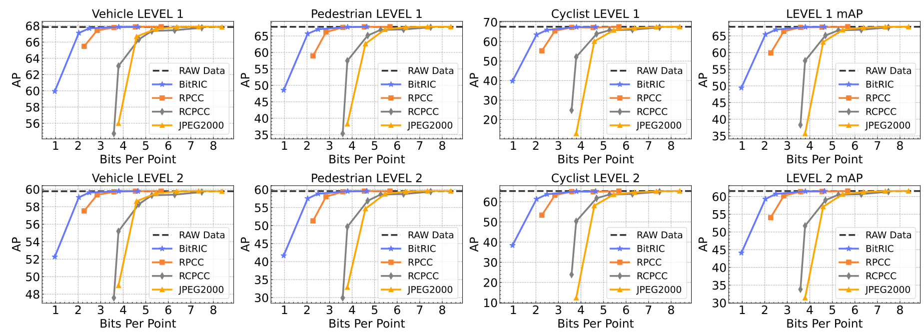

Fig. 6: Quantitative results of 3D object detection on the Waymo Open Dataset. The CenterPoint [39] detector is trained on raw range images and tested on compressed range images. Average Precision (AP) is reported for both "LEVEL 1" and "LEVEL 2" difficulties on the validation set.

Table 3: Contribution of each component in the contextual backbone. BD-BR (D1/D2) is measured relative to the JPEG2000 anchor. Complexity metrics (Params and FLOPs) are evaluated per bitplane.

Range Embedding	Positional Embedding	Horizontal Folding	U-Net-based Aggregation	BD-BR D1 ↓	BD-BR D2 ↓	Params. ↓	FLOPs ↓
✗	✓	✓	✓	-16.26%	-21.03%	1.76M	27.78G
✓	✗	✓	✓	-60.37%	-60.21%	4.42M	27.78G
✓	✓	✗	✓	-61.43%	-61.33%	5.93M	144.55G
✓	✓	✓	✗	-45.06%	-46.39%	2.92M	9.90G
✓	✓	✓	✓	-61.73%	-61.52%	4.42M	27.78G

attributed to two key advantages: (i) the hierarchical bitplane decomposition preserves global geometric integrity across bit-depths, and (ii) the ORM effectively mitigates truncation-induced information loss and compensates for the loss of high-frequency spatial details essential for detection.

4.5 Ablation Study

We conduct ablation studies to evaluate the contribution of each key module. All evaluations in this section are performed on the Waymo Open Dataset.

Contextual Backbone Components. To evaluate the effectiveness of the proposed contextual backbone, we conduct a systematic ablation study of its key components, as reported in Tab. 3. The Range Embedding proves most critical; its removal causes a severe performance drop because the model must then rely on positional cues alone to estimate bitplane distributions. The inclusion of Positional Embedding provides consistent performance gains with negligible computational overhead. Disabling Horizontal Folding increases FLOPs by over 5× (27.78G vs. 144.55G), confirming its essential role in maintaining efficiency without sacrificing reconstruction quality.

Table 4: Impact of the folding ratio r.

Ratio	BD-BR		Params. ↓	FLOPs ↓
	D1 ↓	D2 ↓		
1	-61.43%	-61.33%	5.93M	144.55G
2	-61.70%	-61.55%	4.71M	58.83G
4	-61.73%	-61.52%	4.42M	27.78G
8	-54.83%	-54.29%	4.38M	15.21G

Table 5: Ablation of estimation stages.

Stage	BD-BR		Params. ↓	FLOPs ↓
	D1 ↓	D2 ↓		
1	-24.27%	-25.12%	4.26M	21.55G
2	-50.25%	-50.33%	4.34M	24.66G
3	-61.73%	-61.52%	4.42M	27.78G
4	-61.90%	-61.68%	4.52M	31.64G

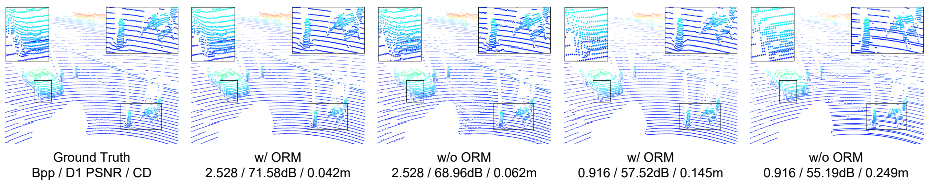

Fig. 7: Impact of the introduced one-step refinement. Visual comparisons are presented at two representative bitrate points. Please zoom in for a better view.

Folding Axis. Range images exhibit a significantly higher horizontal resolution than vertical resolution (e.g., 64 × 2650), which implies stronger spatial correlations along the horizontal axis. To investigate this, we conduct an experiment applying the folding mechanism along the vertical axis. The results show that the BD-BR savings degrade from -61.73% to -50.28% for the D1 metric, and from -61.52% to -50.56% for the D2 metric, thereby confirming the superiority of the horizontal folding mechanism.

Horizontal Folding Ratio. We further investigate the impact of the spatial-to-channel downsampling ratio r in the HSCM. As shown in Tab. 4, setting $r = 1$ (*i.e.*, no folding) incurs a prohibitive computational burden. Increasing r from 1 to 4 results in no significant performance degradation (marginal changes are likely due to training randomness), while yielding a substantial reduction in complexity. However, further increasing r to 8 leads to a marked decline in compression performance. We believe that excessive spatial folding destroys fine-grained local structures essential for the prediction of subsequent bitplanes. Consequently, $r = 4$ is adopted as the default configuration.

Probability Estimation Stages. Table 5 details the ablation study on the cascaded estimation stages within the PEM. A 1-stage approach, which assumes spatial independence within the bitplane, yields a heavily suboptimal BD-BR D1 of -24.27%. Progressively conditioning predictions on previously decoded neighbors via 2- and 3-stage partitions substantially improves the compression gain to -50.25% and -61.73%, respectively. Since a 4-stage partition offers only marginal improvements at a higher computational cost, the 3-stage design is selected to strike an optimal balance between compression ratio and model complexity.

One-Step Refinement. Integrating the ORM consistently enhances reconstruction performance. Qualitative results in Fig. 7 demonstrate that the ORM recovers more precise vehicle contours and roadblock details than the baseline at the same bitrates. Furthermore, it introduces negligible decoding overhead

(2ms on average) because it is invoked only once at the final scale, providing a cost-effective solution for recovering fine-grained geometric details.

Zero-Shot Validation. To further validate the generalization capability of BitRIC, we evaluated the model trained on WOD directly on the SemanticKITTI dataset. Experimental results show that the WOD-trained model achieves a -56.93% BD-BR (D1) relative to the JPEG2000 anchor on SemanticKITTI (compared to 69.92% in-domain), thereby demonstrating the robust generalizability of the model.

Please refer to our supplementary material for more details and discussion.

5 Limitation

Despite the promising performance of BitRIC, several limitations remain. First, BitRIC is currently less competitive than voxel-based approaches (*e.g.*, G-PCC [12] and RENO [40]) at extremely low bitrates (*e.g.*, <1 Bpp). This performance gap stems from structural differences, where voxel-based octrees efficiently downsample point clouds through spatial partitioning. Second, the latency in the lossless mode poses a bottleneck for real-time deployment, requiring further optimizations such as model quantization, which we defer to future research.

6 Conclusion

This paper introduces BitRIC, a neural framework for LiDAR range image compression based on hierarchical bitplane decomposition. The dual-path architecture, incorporating the Probability Estimation Module (PEM) and the One-Step Refinement Module (ORM), enables high-performance lossless and lossy coding within a single model. Experimental results demonstrate that BitRIC significantly outperforms previous methods, achieving significant BD-BR savings and maintaining high fidelity for downstream tasks (*e.g.*, 3D object detection). Coupled with its real-time processing capability, BitRIC offers a compelling solution for efficient LiDAR data compression in practical applications. Future work will focus on exploiting temporal redundancies and model quantization to further optimize streaming performance.

Acknowledgements

This work was supported by the Key Program of the National Natural Science Foundation of China (Grant No. 62431011).

References

1. Abbasi, R., Bashir, A.K., Alyamani, H.J., Amin, F., Doh, J., Chen, J.: Lidar point cloud compression, processing and learning for autonomous driving. IEEE Transactions on Intelligent Transportation Systems **24**(1), 962–979 (2022)

2. Beemelmanns, T., Tao, Y., Lampe, B., Reiher, L., van Kempen, R., Woopen, T., Eckstein, L.: 3d point cloud compression with recurrent neural network and image compression methods. In: 2022 IEEE Intelligent Vehicles Symposium (IV). pp. 345–351. IEEE (2022)
3. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9297–9307 (2019)
4. Bjontegaard, G.: Calculation of average psnr differences between rd-curves. ITU SG16 Doc. VCEG-M33 (2001)
5. Caillet, P., Dupuis, Y.: Efficient lidar data compression for embedded v2i or v2v data handling. arXiv preprint arXiv:1904.05649 (2019)
6. Cao, Y., Wang, Y., Chen, H.: Real-time lidar point cloud compression and transmission for resource-constrained robots. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 12789–12795. IEEE (2025)
7. Feng, Y., Liu, S., Zhu, Y.: Real-time spatio-temporal lidar point cloud compression. In: 2020 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 10766–10773. IEEE (2020)
8. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
9. Gao, W., Zhang, J., Zhao, M., Zhang, Z., Kong, S., Ghaffari, M., Song, D., Xu, C., Kong, H.: Super lidar intensity for robotic perception. IEEE Robotics and Automation Letters (2026)
10. Gao, Y., Zhang, P., Wang, X.: Lossy lidar point cloud compression via cylindrical 3d convolution networks. In: 2023 IEEE International Conference on Image Processing (ICIP). pp. 3508–3512 (2023). `https://doi.org/10.1109/ICIP49359.2023.10222471`
11. Houshiar, H., Nüchter, A.: 3d point cloud compression using conventional image compression for efficient data transmission. In: 2015 XXV international conference on information, communication and automation technologies (ICAT). pp. 1–8. IEEE (2015)
12. ISO/IEC JTC1/SC29/WG7: G-pcc 2nd edition codec description. Output Document N00575, International Organization for Standardization (May 2023), 142th MPEG meeting, Antalya, Turkey
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Kong, L., Liu, Y., Chen, R., Ma, Y., Zhu, X., Li, Y., Hou, Y., Qiao, Y., Liu, Z.: Rethinking range view representation for lidar segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 228–240 (2023)
15. Kong, L., Xu, X., Ren, J., Zhang, W., Pan, L., Chen, K., Ooi, W.T., Liu, Z.: Multimodal data-efficient 3d scene understanding for autonomous driving. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(5), 3748–3765 (2025)
16. Kuwabara, A., Kato, S., Koike-Akino, T., Fujihashi, T.: Range image-based implicit neural compression for lidar point clouds. IEEE Access **14**, 10262–10275 (2026)
17. Lawrence Livermore National Laboratory: FPZIP Version 1.3.0. `https://computing.llnl.gov/projects/fpzip` (Dec 2019)
18. Li, J., Liu, Z., Xu, X., Qin, X., Liu, J., Yuan, S., Fang, X., Xie, L.: Limo-calib: Onsite fast lidar-motor calibration for quadruped robot-based panoramic 3d sensing

system. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8757–8764. IEEE (2025)
19. Li, X., Shang, Y., Hua, B., Yu, R., He, Y.: Lidar intensity correction for road marking detection. Optics and Lasers in Engineering **160**, 107240 (2023)
20. Li, X., Fan, B., Tian, J., Fan, H.: Gafusion: Adaptive fusing lidar and camera with multiple guidance for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21209–21218 (2024)
21. Liang, D., Zhang, C., Zhang, P., Liu, S., Li, H., Niu, S., Rao, R.Z., Zhao, L., Chen, X., Li, H., et al.: Evolution of laser technology for automotive lidar, an industrial viewpoint. nature communications **15**(1), 7660 (2024)
22. Liu, B., Ma, Y., Li, L., Liu, D., Li, Z., Li, H.: Next bit prediction: A unified lossless and lossy point cloud geometry compression framework. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–18 (2026). `https://doi.org/10.1109/TPAMI.2025.3650590`
23. Liu, C.S., Yeh, J.F., Hsu, H., Su, H.T., Lee, M.S., Hsu, W.H.: Bird-pcc: Bi-directional range image-based deep lidar point cloud compression. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2023)
24. Liu, K., Zheng, Q., Tao, K., Li, Z., Qin, H., Li, W., Guo, Y., Liu, X., Kong, L., Chen, G., et al.: Low-bit model quantization for deep neural networks: A survey. arXiv preprint arXiv:2505.05530 (2025)
25. Luo, A., Song, L., Nonaka, K., Unno, K., Sun, H., Goto, M., Katto, J.: Scp: Spherical-coordinate-based learned point cloud compression. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 3954–3962 (2024)
26. Luo, K.Z., Dao, M.Q., Liu, Z., Campbell, M., Chao, W.L., Weinberger, K.Q., Malis, E., Fremont, V., Hariharan, B., Shan, M., et al.: Mixed signals: A diverse point cloud dataset for heterogeneous lidar v2x collaboration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 28763–28773 (2025)
27. McDermott, M., Rife, J.: Correcting motion distortion for lidar scan-to-map registration. IEEE Robotics and Automation Letters **9**(2), 1516–1523 (2023)
28. Milioto, A., Vizzo, I., Behley, J., Stachniss, C.: Rangenet++: Fast and accurate lidar semantic segmentation. In: 2019 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 4213–4220. IEEE (2019)
29. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020)
30. Taubman, D.S., Marcellin, M.W.: Jpeg2000: Standard for interactive imaging. Proceedings of the IEEE **90**(8), 1336–1357 (2002)
31. Team, O.D.: Openpcdet: An open-source toolbox for 3d object detection from point clouds. `https://github.com/open-mmlab/OpenPCDet` (2020)
32. Tu, C., Takeuchi, E., Carballo, A., Takeda, K.: Point cloud compression for 3d lidar sensor using recurrent neural network with residual blocks. In: 2019 international conference on robotics and automation (ICRA). pp. 3274–3280. IEEE (2019)
33. Tu, C., Takeuchi, E., Carballo, A., Takeda, K.: Real-time streaming point cloud compression for 3d lidar sensor using u-net. IEEE Access **7**, 113616–113625 (2019)
34. Tu, C., Takeuchi, E., Miyajima, C., Takeda, K.: Compressing continuous point cloud data using image compression methods. In: 2016 IEEE 19th international conference on intelligent transportation systems (ITSC). pp. 1712–1719. IEEE (2016)

35. Van Beek, P.: Image-based compression of lidar sensor data. Electronic Imaging **31**, 1–7 (2019)
36. Wang, S., Jiao, J., Cai, P., Wang, L.: R-pcc: A baseline for range image-based point cloud compression. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 10055–10061. IEEE (2022)
37. Xue, R., Li, J., Chen, T., Ding, D., Cao, X., Ma, Z.: Neri: Implicit neural representation of lidar point cloud using range image sequence. In: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8020–8024. IEEE (2024)
38. Yang, B., Li, Z., Li, W., Cai, Z., Wen, C., Zang, Y., Muller, M., Wang, C.: Lisa: Lidar localization with semantic awareness. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15271–15280 (2024)
39. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11784–11793 (2021)
40. You, K., Chen, T., Ding, D., Asif, M.S., Ma, Z.: Reno: Real-time neural compression for 3d lidar point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22172–22181 (2025)
41. Zhao, J., Wu, Y., Deng, R., Xu, S., Gao, J., Burke, A.: A survey of autonomous driving from a deep learning perspective. ACM Computing Surveys **57**(10), 1–60 (2025)
42. Zhao, L., Ma, K.K., Liu, Z., Yin, Q., Chen, J.: 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 (2022)
43. Zhou, X., Qi, C.R., Zhou, Y., Anguelov, D.: Riddle: Lidar data compression with range image deep delta encoding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17212–17221 (2022)