

AdaDPCC: Adaptive Rate Control and Rate-Distortion-Complexity Optimization for Dynamic Point Cloud Compression

Chenhao Zhang¹, Wei Gao^{1,2*}

¹Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology, Shenzhen Graduate School, Peking University, Shenzhen 518055, China

²Peng Cheng Laboratory, Shenzhen 518055, China
chenhaozhang@stu.pku.edu.cn, gaowei262@pku.edu.cn

Abstract

Dynamic point cloud compression (DPCC) is crucial in applications like autonomous driving and AR/VR. Current compression methods face challenges with complexity management and rate control. This paper introduces a novel dynamic coding framework that supports variable bitrate and computational complexities. Our approach includes a slimmable framework with multiple coding routes, allowing for efficient Rate-Distortion-Complexity Optimization (RDCO) within a single model. To address data sparsity in inter-frame prediction, we propose the coarse-to-fine motion estimation and compensation module that deconstructs geometric information while expanding the perceptive field. Additionally, we propose a precise rate control module that content-adaptively navigates point cloud frames through various coding routes to meet target bitrates. The experimental results demonstrate that our approach reduces the average BD-Rate by 5.81% and improves the BD-PSNR by 0.42 dB compared to the state-of-the-art method, while keeping the average bitrate error at 0.40%. Moreover, the average coding time is reduced by up to 44.6% compared to D-DPCC, underscoring its efficiency in real-time and bitrate-constrained DPCC scenarios.

Introduction

Dynamic point clouds, composed of sequences of point cloud frames, play crucial roles in applications such as robot sensing, autonomous driving, and AR/VR. These immersive technologies demand the real-time rendering of vast amounts of 3D data, posing significant challenges for data storage and transmission. To facilitate seamless and high-quality VR experiences, a robust yet flexible dynamic point cloud compression (DPCC) algorithm is indispensable, ensuring adaptable solutions under constraints of bandwidth and computational resources.

According to source coding theory (Goyal 2001), the essence of compression is the maximal elimination of source redundancy. In this context, existing DPCC efforts primarily focus on reducing spatial and temporal redundancies. To exploit spatial redundancy, current methods mainly utilize Variational Autoencoders to progressively downsample and aggregate spatial information into a latent space (Yan

et al. 2019; Wang et al. 2021a; Fan et al. 2022; Xia et al. 2023). This results in a less-correlated latent representation, which is subsequently quantized and entropy-coded using established probability models (Minnen, Ballé, and Toderici 2018; Ballé et al. 2018). A significant limitation of these approaches is that they are optimized to a specific Rate-Distortion (RD) trade-off within a single model, limiting their flexibility in various bitrate scenarios. Conversely, exploiting temporal redundancy depends heavily on the precision of inter-frame coding. Current strategies typically involve motion estimation and KNN-based motion compensation within latent space (Fan et al. 2022; Xia et al. 2023; Pan et al. 2024). However, the high sparsity of downsampled points obscures the correspondence between points across frames, which in turn limits the perceptive field of KNN-based methods. The coding time is also significantly extended due to the computationally intensive nature of the KNN algorithm. Furthermore, existing methods often fail to provide precise rate control, lacking both accurate rate estimation and variable rate solutions, which are crucial for adjusting coding quality under diverse bandwidth constraints.

To address these challenges, this paper introduces a novel dynamic coding framework tailored for varying bitrates and computational complexities. We propose a dynamic Variational Autoencoder-based framework to efficiently exploit spatial redundancy, which includes multiple coding routes. Each route takes up a partial of the overall architectural computational complexity, achieving a distinct RD trade-off. Through joint training of these routes, optimal RD performance is attained at each complexity level, thereby facilitating RDCO within a single model. For temporal redundancy, we present a coarse-to-fine inter-prediction module that initially estimates motion embeddings and subsequently deconstructs geometric information from the latent space to provide anchors for precise motion compensation. Compared to KNN-based methods, our approach significantly reduces inter-prediction time while expanding the receptive field, leading to improved RD performance. Additionally, to achieve low-latency yet precise rate control, we introduce a lightweight rate control module that predicts the bitrate for each route and selects the optimal coding route using a sliding window and bit allocation algorithm.

The contributions of our work are as follows:

- To achieve RDCO in the realm of DPCC, our method

*Corresponding Author.
Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

dynamically allocates computational complexity, maintaining superior RD performance. This flexibility allows for variable bitrate and adaptive computational resource allocation under bandwidth constraints.

- To address data sparsity in inter-frame prediction, we introduce coarse-to-fine motion estimation and compensation, performing block-wise and voxel-wise motion estimation and anchor-based motion compensation.
- To ensure precise rate control, the proposed method adaptively evaluates the bitrate of each route based on content and determines the optimal coding route through the sliding window and bit allocation algorithm.

Related Work

Learning-based Point Cloud Compression

The primary objective of learning-based PCC is to minimize the RD loss as:

$$L = R + \lambda D \quad (1)$$

$$= -\mathbb{E}(\log_2 P_{\tilde{y}}(\tilde{y})) + \lambda d(x, \hat{x})$$

where λ is the Lagrange multiplier, x is the input frame and \hat{x} is the reconstructed frame, $P_{\tilde{y}}(\cdot)$ is the probability distribution of quantized latent representation \tilde{y} and $d(\cdot, \cdot)$ is the criterion to measure the similarity between two frames.

Learning-based PCC can be classified into point-based, voxel-based, and octree-based methods, each distinguished by the data format employed during the coding process.

Point-based methods utilize PointNet++-based auto-encoders to process point cloud coordinates and feature extraction. Huang and Liu (2019) implemented Farthest Point Sampling (FPS) for downsampling coordinates and employed entropy coding for latent features. To mitigate the computational demands of FPS, Gao et al. (2021) introduced Neural Graph Sampling with attention-based mechanisms on local graphs. He et al. (2022) used sub-point convolution to preserve density information during upsampling, and Li et al. (2024) innovated with a learnable sampler and Wasserstein distance for improved distribution fidelity. Despite their efficiency, point-based methods face challenges in memory management for large point clouds.

Voxel-based methods transform raw coordinates into gridded voxels, utilizing sparse convolution (Choy, Gwak, and Savarese 2019; Tang et al. 2023) to optimize memory usage. Quach, Valenzise, and Dufaux (2019) introduced 3D voxelized convolution within an auto-encoder framework, enhanced by a hyperprior framework in subsequent versions (Quach, Valenzise, and Dufaux 2020) to better explore spatial redundancy. Wang et al. (2021b) incorporated Voxception-ResNet into their auto-encoder, with Wang et al. (2021a) adding multi-scale upsampling to the decoder for improved transformation capabilities. Wang et al. (2023b) developed a unified framework for lossy and lossless compression, segmenting the coding process into scalable levels. Despite its effectiveness, the use of sparse convolution is restricted in handling context features, especially in sparse distributions from LiDAR scans.

Octree-based methods leverage the efficient octree structure to extract contextual information effectively. Huang

et al. (2020) initiated entropy coding of occupancy codes using ancestor nodes. Que, Lu, and Xu (2021) refined the modeling of occupancy symbol probability distributions by leveraging local voxel contexts. Fu et al. (2022) addressed gaps in resolution by integrating contexts from both ancestor and sibling nodes to minimize redundancy. To overcome the bottlenecks in serial decoding, Song et al. (2023) introduced a hierarchical attention model, enhancing throughput while maintaining extensive context capture.

Dynamic Point Cloud Compression

DPCC primarily targets RD optimization across a Group of Frames (GoF), leveraging inter-frame correlation to enhance temporal context. Rule-based methods have been pivotal, with V-PCC (Schwarz et al. 2019) encoding 3D frames onto 2D planes using traditional video coding techniques. Recent learning-based advancements have excelled in non-linear transform coding (Ballé et al. 2021) and probability estimation accuracy. Specifically, Akhtar, Li, and Auwera (2024) integrated temporal information through sparse convolution and residual coding, resulting in superior RD metrics compared to V-PCC. However, this approach is limited by lacking explicit motion prediction. Fan et al. (2022) enhanced motion estimation with multi-scale modules, and Xia et al. (2023) innovated with a KNN-attention block-matching (KABM) module for precise feature aggregation. Furthermore, Pan et al. (2024) divided point cloud frames into patches to better explore inter-frame redundancy, setting new RD performance benchmarks. Despite these advancements, the computational demand of the KNN method limits the K value in inter-prediction, thereby impacting local context capture. Additionally, the effectiveness of residual coding relies heavily on the accuracy of inter-prediction, underscoring the need for richer context exploitation.

Dynamic Neural Network

Dynamic Neural Network (DNN) is a novel category that adapts its architecture (Veit and Belongie 2020; Yang et al. 2024; Zhang and Gao 2024) and parameters (Wang et al. 2023c; Qi et al. 2023) based on varying input data, making them highly suitable for learning-based compression. Addressing the demands for adaptive coding, Yang et al. (2021) introduced a slimmable auto-encoder to execute distinct RD trade-offs at various scales. Additionally, Tao et al. (2023) segmented input images into patches, dynamically assigning model capacities to match the coding complexities of each patch. Hu and Xu (2023) further leveraged a slimmable decoder to adjust decoding complexity. Despite these innovations, developing an efficient DNN-based coding network for precise rate control remains a major challenge, highlighting the need for advanced DNN solutions.

Methodology

Overview

To achieve RDCO and precise rate control, we propose a dynamic coding network with multiple coding routes. Specifically, to approach the target bitrate R_{tar} , the rate control module first estimates the coded bitrates and determines the

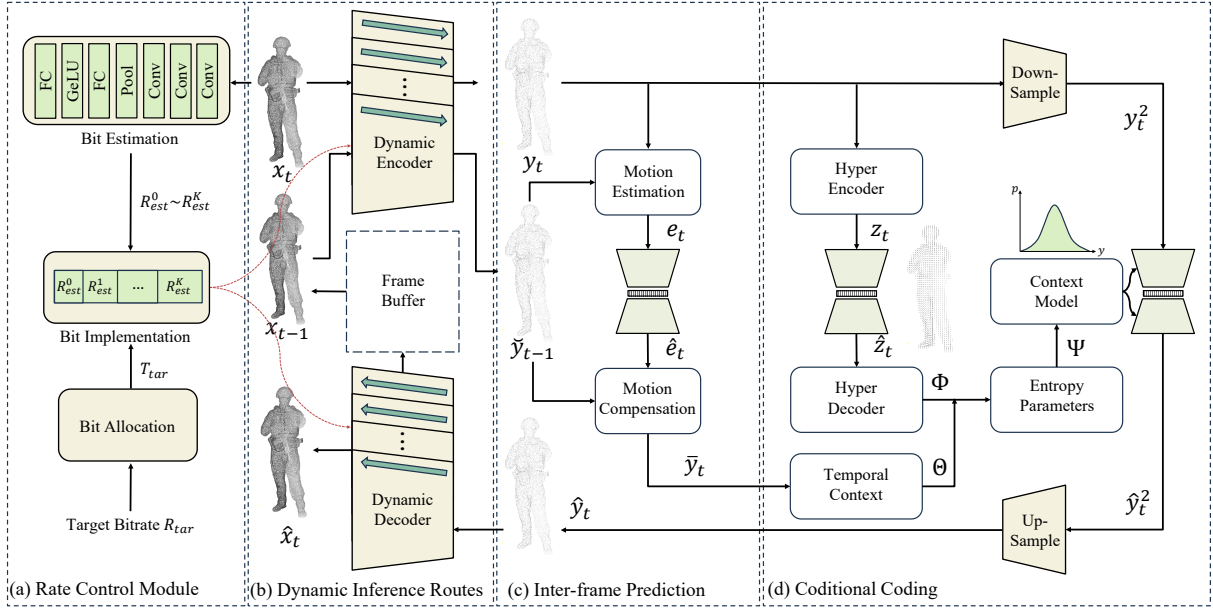


Figure 1: Overview of the proposed method: (a) Rate Control Module. It content-adaptively allocates coding routes for point cloud frames to meet target bitrates (b) Dynamic Inference Routes. The Dynamic Encoder/Decoder contains K routes, each taking up partial of the total complexity and navigating to a unique RD trade-off. (c) Inter-frame Prediction. It exploits inter-frame redundancy with Motion Estimation and Motion Compensation modules. (d) Conditional Coding. It merges hyper-prior and temporal contexts to model the distribution of downsampled latent representation.

optimal route k for the current frame x_t . This frame is then encoded into latent representation y_t via the k -th route of the dynamic encoder. Simultaneously, the reference frame \hat{x}_{t-1} is encoded through the same route to generate \hat{y}_{t-1} . Coarse-to-fine motion estimation and compensation are applied between y_t and \hat{y}_{t-1} , followed by conditional coding with the context of the compensated latent variable \bar{y}_t and the hyper-prior \hat{z}_t . Finally, The reconstructed frame \hat{x}_t is decoded via the same k -th route and buffered for subsequent inter-frame prediction. The overall architecture is shown in Figure 1.

Dynamic Inference Routes

It is noted that during the encoding of diverse point cloud contents, the neural network’s active regions correlate with the contents’ complexity, resulting in different active regions leading to varying bitrates. Therefore, our framework incorporates multiple coding routes, each contributing differently to computational complexity and resulting in distinct RD trade-offs. Specifically, the Dynamic Encoder $g_a(\cdot; \theta)$ and Dynamic Decoder $g_s(\cdot; \phi)$ utilize slimmable operators to dynamically adjust the channel dimensions of the input features, allowing the network to efficiently manage varying computational demands. These operators enable the division of the overall coding network (supernet) into several overlapping sub-networks (subnets), each handling a portion of the total computational load, as shown in Figure 2.

In the encoding process, the current point cloud frame $x_t = \{C_t, F_t\}$, where $C_t \in \mathbb{N}^{N \times 3}$ represents spatial coordinates and $F_t \in \mathbb{R}^{N \times D}$ denotes associated features with D dimensions, is transformed into a D_k -dimensional latent

representation $y_t = \{C'_t, F'_t\}$ via the selected route k :

$$y_t = g_a(x_t; \theta^{\leq k}) = \{C'_t, F'_t\}, \quad (2)$$

where $\theta^{\leq k}$ denotes the parameters for the k -th route of the Encoder $g_a(\cdot; \theta)$. During encoding, the original point clouds are downsampled to sparser coordinates C'_t to encapsulate local geometric details in the feature space, enhancing the information capacity of the latent representation. Subsequently, y_t is entropy coded under temporal and spatial contexts to maximize redundancy reduction, which will be elaborated in subsequent sections.

In the decoding process, the entropy-decoded latent representation \hat{y}_t is transformed back to the reconstructed frame $\hat{x}_t = \{\hat{C}_t, F_t\}$ through the same route k :

$$\hat{x}_t = g_s(\hat{y}_t; \phi^{\leq k}) = \{\hat{C}_t, F_t\}, \quad (3)$$

where $\phi^{\leq k}$ denotes the parameters for the k -th route of the Decoder $g_s(\cdot; \phi)$. Notably, in voxel space, features F_t primarily serve as occupancy indicators and are uniformly set to all-one vectors.

As previously noted, routes with more parameters preserve more local geometric information, enhancing reconstruction quality at the expense of higher bitrate and computational complexity. However, it is inefficient to train each route independently due to the parameter interdependence among routes. To tackle this issue, we employ a joint training strategy, described in Algorithm 1. Initially, a supernet with comprehensive parameters is pre-trained using a high λ to support high bitrate demands. Each subnet is then trained

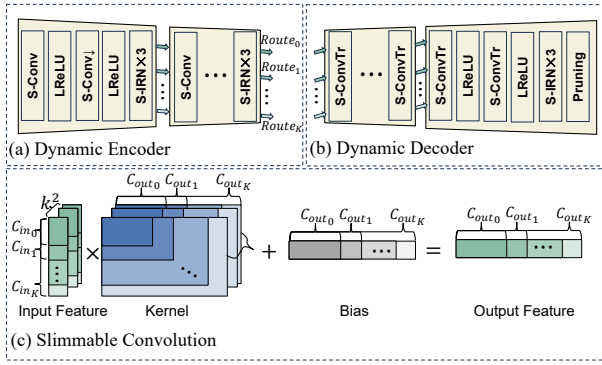


Figure 2: Dynamic inference routes: (a) Dynamic Encoder with K routes, each consists of two downsample blocks. (b) Dynamic Decoder with matching upsample blocks. ‘‘S-X’’ stands for ‘‘Slimmable X’’. (c) Slimmable Convolution transferring variable input channels to output channels.

sequentially with decreasing λ values, using the cumulative RD loss as the joint optimization criterion. Subsequently, the post-training process gradually reduces λ_0 for route 0 to achieve the lowest acceptable bitrate.

Algorithm 1: Joint Routes Training Strategy

- 1: **Input:** training iteration N , model $Model$, number of routes K , λ list $[\lambda_0, \lambda_1, \dots, \lambda_{K-1}]$, train dataset χ_{train} ;
 - 2: pre-train the supernet with λ_{K-1} for maximum bitrate.
 - 3: **for** $j = 1, 2, \dots, N$ **do**
 - 4: $P_{input}, P_{ref} \leftarrow \chi_{train}$
 - 5: **for** $k = K - 2, K - 3, \dots, 0$ **do**
 - 6: $R, D \leftarrow Model(P_{input}, P_{ref})$;
 - 7: $Loss \leftarrow Loss + R + \lambda_k D$;
 - 8: **end for**
 - 9: update $Model$ parameters;
 - 10: **end for**
 - 11: post-train by gradually decreasing λ_0
-

Through joint routes training, the coding routes are collaboratively optimized to achieve optimal RD performance at each level of complexity. This approach allows the model to effectively manage computational complexity corresponding to variable bitrate, thereby supporting efficient RDCO within a single model.

Coarse-to-fine Inter-frame Prediction

The accuracy of inter-frame prediction significantly impacts the quality of temporal context and, consequently, the RD performance. To address the challenges posed by large intervals in down-sampled coordinates and a limited perception field in inter-frame prediction, we introduce the Coarse-to-fine Inter-frame Prediction method, as shown in Figure 3.

Motion estimation. To achieve end-to-end optimization, inter-frame prediction is conducted in feature space. Temporal information from consecutive frames is fused by concatenating $y_t = \{C'_t, F'_t\}$ and $\check{y}_{t-1} = \{C'_{t-1}, F'_{t-1}\}$ as:

$$C^{cat} = C'_t \cup C'_{t-1}, \quad (4)$$

$$F^{cat} = \begin{cases} F'_{t,m} \oplus F'_{t-1,m} & m \in C'_t \cap C'_{t-1} \\ F'_{t,m} \oplus 0 & m \in C'_t, m \notin C'_{t-1} \\ 0 \oplus F'_{t-1,m} & m \notin C'_t, m \in C'_{t-1} \end{cases}, \quad (5)$$

$$y^{cat} = \{C^{cat}, F^{cat}\}, \quad (6)$$

where y^{cat} is the fused frame, C^{cat} and F^{cat} are the corresponding coordinates and features, m represents any single point in C^{cat} , and \oplus denotes channel-wise concatenation. The fused frame y^{cat} is processed by the Motion Estimation module, which initially applies coarse-grained convolutions with the tensor stride of 2. This is followed by fine-grained convolutions with the tensor stride of 1, producing a fine-grained motion embedding e_f . To mitigate information loss from network depth, a residual branch directly integrates residual details with e_f . To ensure precise alignment with y_t , the coordinates of e_t are pruned to match C'_t , resulting in the final motion embedding $e_t = \{C'_t, m_t\}$. The coordinates of e_t are encoded losslessly, while the 3D motion vector m_t is quantized and entropy encoded into bitstream.

Motion compensation. The large intervals in down-sampled coordinates constrain the perceptive field, limiting the effectiveness of motion compensation. To address this, we introduce a two-shot motion compensation approach that first deconstructs geometric information from the feature space and then fuses neighboring features based on reconstructed anchors. This process begins by quantizing the decoded motion vector \hat{m}_t to a $2 \times$ down-sampled scale:

$$\hat{m}_t^c = \lfloor \hat{m}_t / 2 \rfloor \times 2, \quad (7)$$

where \hat{m}_t^c represents the coarse-grained motion vector that captures block-wise motion vector. Subsequently, we warp the coordinates C'_t and combine them with C'_{t-1} . To extract and deconstruct the geometric information from \check{y}_{t-1} into the warped voxels, the combined voxels are processed through the geometric deconstruction module $g_{gd}(\cdot)$:

$$y_t^{anchor} = g_{gd}(\{(\hat{m}_t^c + C'_t), 0\} \cup \{C'_{t-1}, F'_{t-1}\}), \quad (8)$$

where $y_t^{anchor} = \{C_t^{anchor}, F_t^{anchor}\}$ serves as the anchor, integrating geometric details from both voxel and feature spaces, thereby providing a rich and precise contextual anchor. For finer motion adjustments, we generate a fine-grained motion vector:

$$\hat{m}_t^f = \lfloor \hat{m}_t / 1 \rfloor \times 1, \quad (9)$$

where \hat{m}_t^f denotes fine-grained motion depicting voxel-wise motion vector. We then warp the coordinates of C'_t and merge them with C_t^{anchor} . These merged voxels are subsequently refined by the feature aggregation module $g_{fa}(\cdot)$ to align features from y_t^{anchor} into the compensated latent representation \bar{y}_t :

$$\bar{y}_t = g_{fa}(\{(\hat{m}_t^f + C'_t), 0\} \cup y_t^{anchor}), \quad (10)$$

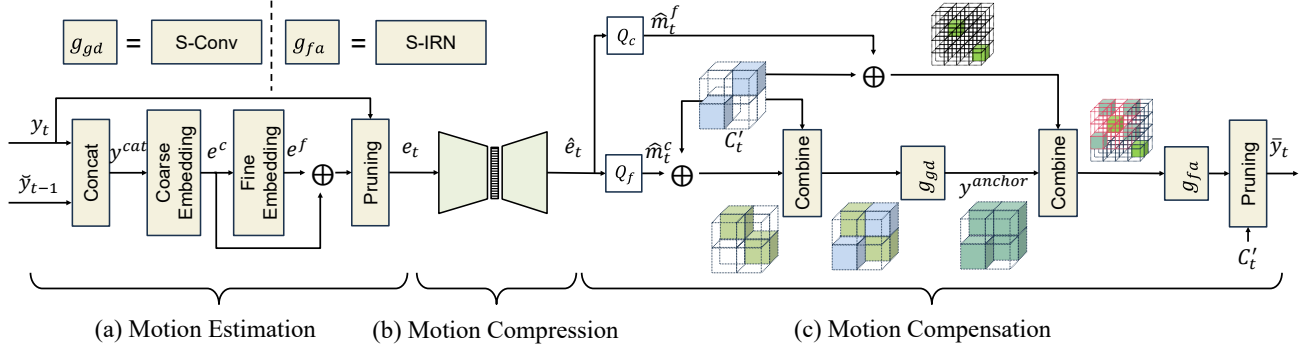


Figure 3: Illustration of the coarse-to-fine inter-frame prediction.

where \bar{y}_t represents the predicted current frame in latent space, serving as the temporal context for conditional inter-frame coding. Through coarse-to-fine motion compensation, geometric information is reconstructed from the feature space, offering anchors that greatly enhance the accuracy of motion compensation.

Conditional Inter-frame Coding

To fully exploit geometric redundancy, we implement conditional inter-frame coding that first extracts context information from both temporal and spatial aspects. The overall context is generated as:

$$\Psi = g_{ep}(\Theta, \Phi; \phi_{ep}), \quad (11)$$

where Θ represents the temporal context, Φ denotes the hyper-prior context, and Ψ serves as the overall context in conditional coding, with $g_{ep}(\cdot; \phi_{ep})$ being the entropy parameters network. The conditional probability $p_{\bar{y}|\Psi}$ is modeled as a factorized Gaussian distribution integrated with soft quantization noise:

$$p_{\bar{y}|\Psi}(\bar{y}|\Psi) = \prod_{i=0} (N(\mu_i, \sigma_i^2) * U(-\frac{1}{2}, \frac{1}{2}))(\bar{y}_{t,i}), \quad (12)$$

where $U(-\frac{1}{2}, \frac{1}{2})$ indicates the uniform distribution of soft quantization noise. Finally, the RD loss for the k -th route is:

$$\begin{aligned} \mathcal{L}_k &= R_m + R_z + R_y + \lambda_k D \\ &= -\mathbb{E}(\log_2 P_{\tilde{m}}(\tilde{m})) - \mathbb{E}(\log_2 P_{\tilde{z}}(\tilde{z})) \\ &\quad - \mathbb{E}(\sum_{i=1} \log_2(p_{\bar{y}_i}(\tilde{y}_i|\Psi))) + \lambda_k d(x, \hat{x}), \end{aligned} \quad (13)$$

where R_m , R_z , and R_y denote the rate losses for motion embedding, hyper-prior, and latent representation, respectively. Through end-to-end training, R_m and R_z contribute only a marginal portion to the final bitstream yet significantly enhance RD performance.

Rate Control Module

To achieve precise frame-level rate control, each frame's coding route is carefully selected to match the target bitrate R_{tar} . It is well-recognized that a frame's bitrate is closely

linked to its spatial distribution and geometric representation. In the proposed framework, we integrate three key components to ensure accurate rate control: the Rate Estimation module, the Bit Allocation module, and the Bit Implementation module.

First, the Rate Estimation module estimates the bitrates $R_{est}^0 \sim R_{est}^K$ for each coding route. Then, the Bit Allocation module employs a sliding window algorithm for frame-level bitrate allocation, determining the target bitrate T_{tar} for the current frame:

$$T_{tar} = \frac{R_{tar} \times (N_c + SW) - R_c}{SW}, \quad (14)$$

where T_{tar} represents the target bitrate for the current frame, N_c is the number of frames that have already been encoded, R_c refers to the bits used by these frames, and SW is the length of the sliding window. This algorithm aims for bitrate stability across SW frames, promoting consistent video quality. Notably, at the start of a GoF, the "I" frame is typically allocated a higher bitrate to enhance RD performance in subsequent frames, deviating from the regular allocation strategy to mitigate cumulative errors.

Finally, the Bit Implementation module identifies current route i^* based on the current bit consumption. If the allocated bits exceed the consumed bits, the module selects the route with an estimated bitrate slightly above T_{tar} , or the highest available bitrate route if no such match is found. On the other hand, if the allocated bits are less than the consumed bits, the module chooses a route with an estimated bitrate slightly below T_{tar} , or the lowest available bitrate route if no suitable option is available. The bit implementation strategy is described as follows:

$$i^* = \begin{cases} \arg \min_i (R_{est}^i - T_{tar}), & \text{if } R_{tar} \times N_c > R_c \\ \text{s.t. } R_{est}^i > T_{tar} \\ \arg \min_i (T_{tar} - R_{est}^i), & \text{if } R_{tar} \times N_c < R_c \\ \text{s.t. } R_{est}^i < T_{tar} \end{cases} \quad (15)$$

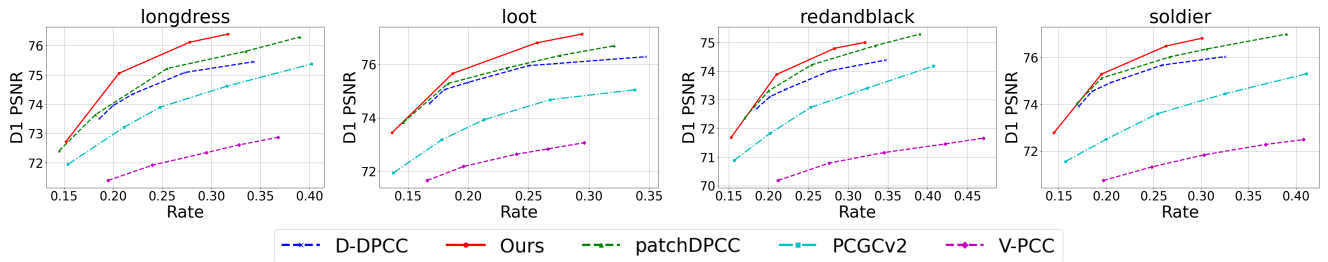


Figure 4: D1-PSNR RD curves of different methods on 8iVFB dataset.

Datasets	D1 BD-Rate(%) / BD-PSNR(dB)			
	D-DPCC	patchDPCC	PCGCv2	V-PCC
Longdress	-14.324/1.016	-9.435/0.623	-26.254/1.758	-55.170/3.750
Loot	-11.071/0.700	-3.720/0.427	-31.723/1.932	-59.373/3.790
Redandblack	-10.021/0.729	-4.145/0.307	-24.806/1.704	-68.133/3.910
Soldier	-9.485/0.683	-5.941/0.338	-59.764/2.747	-67.082/4.935
Average	-11.225/0.782	-5.810/0.423	-35.636/2.035	-62.439/4.096

Table 1: RD performance comparison of different methods on 8iVFB dataset.

Methods	Coding Time (s)
D-DPCC	1.28/1.16
PCGCv2	1.09/0.84
Ours(Route 3)	1.07/1.16
Ours(Route 2)	0.85/0.81
Ours(Route 1)	0.77/0.75
Ours(Route 0)	0.68/0.67

Table 2: Average coding time comparison (encoding time/decoding time) on 8iVFB dataset.

Experiments

Implementation Details

Datasets. We train our model using the OwlII dataset (Cao et al. 2018), which comprises four sequences totaling 2400 frames, with each sequence lasting 20 seconds at 30 frames per second (FPS). To optimize training efficiency, we quantize the original 11-bit precision point cloud data to 10-bit. For model evaluation, we utilize the MPEG 8iVFB dataset (d’Eon et al. 2017), consisting of four sequences with 1200 frames, where each sequence spans 10 seconds at 30 FPS. We adhere to the MPEG common test condition (CTC), setting the GoF size to 32. The first frame of each GoF is designated as an “I” frame and encoded by PCGCv2 (Wang et al. 2021a) with the same λ , while subsequent frames are labeled as “P” frames and encoded by the proposed method.

Network configuration. Through numerous trials, we finalized our network configuration to include four coding routes ($K = 4$), with a Lagrange multiplier list set to [3, 7, 10, 20]. The training strategy follows the protocol established in Algorithm 1. The training is conducted on an NVIDIA A100 GPU, lasting for 100 epochs with the batch size of 2.

Evaluation metrics. We measure bitrate in bits per point

(bpp), calculated by dividing the total bitstream size by the number of input points. Distortion is assessed using point-to-point (D1) PSNR and point-to-plane (D2) PSNR, which quantify the reconstruction accuracy compared to the original point clouds. Rate control accuracy is evaluated by the bitrate error $\Delta R = \frac{|R_{out} - R_{tar}|}{R_{tar}} \times 100\%$, and coding time complexity by the average coding time per frame $T = \frac{T_{total}}{N_{frame}}$.

Benchmark models. To validate the effectiveness of our approach, we compare it against several PCC methods: rule-based V-PCC test model v18 (Schwarz et al. 2019), learning-based D-DPCC (Fan et al. 2022), patchDPCC (Pan et al. 2024), and PCGCv2 (Wang et al. 2021a). Static PCC methods like PCGCv2 encode frames independently. The results for patchDPCC, whose source code is unavailable, are taken directly from the original paper. To ensure fairness, we maintain identical testing conditions to patchDPCC. Additionally, D-DPCC and PCGCv2 are retrained under the same conditions as the proposed method.

Experimental Results

RD performance. Figure 4 shows the RD curves of our method on the 8iVFB dataset, illustrating superior RD performance in a single model compared to benchmark methods. Table 1 further details quantitative results. Our method outperforms rule-based V-PCC, achieving an average BD-Rate reduction of 62.439% and a 4.096 dB improvement in BD-PSNR. Against the advanced learning-based method patchDPCC, our method shows a BD-Rate reduction of 5.81% and a 0.423 dB increase in BD-PSNR.

Coding time. The coding times for various methods are compared in Table 2, all evaluated under the same hardware and software environment using an Nvidia GeForce GTX 3090 GPU with CUDA 11.6. The coding time for V-PCC,

Target bitrate	Bitrate error $\Delta R\%$								Average
	0.15	0.17	0.19	0.21	0.23	0.25	0.27	0.29	
Longdress	0.07	0.25	0.23	0.28	0.33	0.42	0.53	0.32	0.30
Loot	0.44	0.23	1.02	0.27	0.18	0.10	0.01	1.16	0.43
Redandblack	1.12	0.67	1.43	0.21	0.69	0.16	1.00	0.22	0.69
Soldier	0.37	0.42	0.38	0.04	0.06	0.10	0.12	0.03	0.19

Table 3: Bitrate error comparison on 8iVFB dataset.

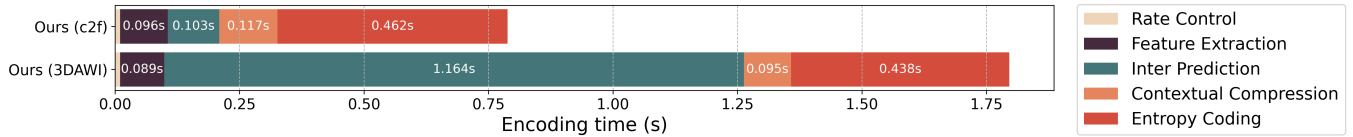


Figure 5: Average encoding time comparison between the proposed coarse-to-fine (c2f) inter-frame prediction and the 3DAWI method on the *Soldier* sequence. Both methods were controlled to achieve 0.20 bpp for fairness.

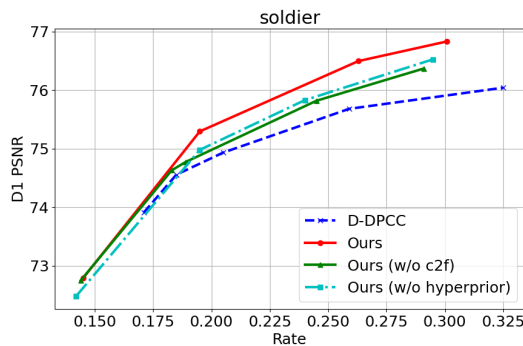


Figure 6: Ablation study on RD performance: comparison of our method with and without the coarse-to-fine (c2f) module and hyperprior context.

which exceeds 50 seconds, is not listed as it operates on a CPU platform, contrasting with the GPU-based environment used for learning-based methods. Our method’s maximum coding time per frame (via Route 3) is approximately 2.23 seconds, which is still 8.2% faster than D-DPCC. As the computational load decreases, the coding time reduces significantly, with a 39.4% reduction from Route 3 to Route 0, and it is 44.6% faster than D-DPCC, demonstrating our method’s efficiency for real-time DPCC applications.

Rate control accuracy. Our rate control module’s accuracy is tested by setting target bitrates from 0.15 to 0.29 bpp in increments of 0.02 bpp. As shown in Table 3, our method consistently achieved close adherence to the set targets, with an average bitrate error of only 0.40%, showcasing its precise rate control capabilities.

Ablation Study

Coding latency. To validate the low coding latency of our proposed coarse-to-fine inter-frame prediction, we conducted an ablation study by replacing it with an alternative inter-frame prediction method, KNN-based 3DAWI (Fan

et al. 2022), and measured the encoding time of each component. Experimental results on the *Soldier* sequence are shown in Figure 5. The decomposed encoding time reveals that the 3DAWI method occupies 64.8% of the total encoding time. In contrast, our proposed coarse-to-fine prediction model requires only 8.8% of the time needed by 3DAWI, accounting for just 13.1% of the total encoding time. This substantial reduction is primarily due to the use of computationally efficient sparse convolution instead of the time-consuming KNN. Additionally, Figure 5 shows that the inference latency of the rate control module is negligible, which is approximately 0.01 seconds, making it an efficient solution for real-time rate control.

RD performance. To demonstrate the RD improvements achieved by the coarse-to-fine inter-frame prediction and hyper-prior context, we conducted an ablation study substituting the former with 3DAWI and removing the latter. Ablation results, shown in Figure 6, indicate that our method’s RD improvement is driven by both components: the coarse-to-fine inter-frame prediction provides a 3.4% BD-Rate reduction, while the hyper-prior context contributes an additional 2.9%. These findings highlight the significant contributions of both components in enhancing RD performance.

Conclusion

In this work, we introduce a novel DPCC framework that achieves variable bitrate and computational complexities with an efficient rate control mechanism. By coarse-to-fine inter-frame prediction, the receptive field is expanded for precise motion estimation and compensation. To meet target bitrates, the lightweight rate control module adaptively navigates point cloud frames through various coding routes, ensuring precise rate control. Experimental results further demonstrate the proposed method’s RDCO effectiveness and rate control accuracy, offering a flexible solution for real-time, bitrate-constrained DPCC applications.

Acknowledgments

This work was supported by The Major Key Project of PCL (PCL2024A02), Natural Science Foundation of China (62271013, 62031013), Guangdong Provincial Key Laboratory of Ultra High Definition Immersive Media Technology (2024B1212010006), Guangdong Province Pearl River Talent Program (2021QN020708), Guangdong Basic and Applied Basic Research Foundation (2024A1515010155), Shenzhen Science and Technology Program (JCYJ20240813160202004, JCYJ20230807120808017).

References

- Ahn, J.; Pang, J.; Lodhi, M. A.; and Tian, D. 2023. DDA-Net: Deep Distribution-Aware Network for Point Cloud Compression. In *IEEE International Symposium on Circuits and Systems*, 1–5.
- Akhtar, A.; Li, Z.; and Auwera, G. V. D. 2024. Inter-Frame Compression for Dynamic Point Cloud Geometry Coding. *IEEE Trans. Image Process.*, 33: 584–594.
- Ballé, J.; Chou, P. A.; Minnen, D.; Singh, S.; Johnston, N.; Agustsson, E.; Hwang, S. J.; and Toderici, G. 2021. Non-linear Transform Coding. *IEEE J. Sel. Top. Signal Process.*, 15(2): 339–353.
- Ballé, J.; Minnen, D.; Singh, S.; Hwang, S. J.; and Johnston, N. 2018. Variational image compression with a scale hyperprior. In *Int. Conf. Learn. Represent.*
- Cao, K.; Xu, Y.; Lu, Y.; and Wen, Z. 2018. OwlII Dynamic Human Mesh Sequence Dataset. Presented at the 122th MPEG Meeting, ISO/IEC JTC1/SC29/WG11 m42816.
- Chou, P. A.; Koroteev, M.; and Krivokuca, M. 2020. A Volumetric Approach to Point Cloud Compression - Part I: Attribute Compression. *IEEE Trans. Image Process.*, 29: 2203–2216.
- Choy, C. B.; Gwak, J.; and Savarese, S. 2019. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 3075–3084.
- 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 Williams, B.; Chen, Y.; and Neville, J., eds., *AAAI*, 470–478.
- d’Eon, E.; Harrison, B.; Myers, T.; and Chou, P. A. 2017. 8i Voxelized Full Bodies: A Voxelized Point Cloud Dataset. ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006.
- Fan, T.; Gao, L.; Xu, Y.; Li, Z.; and Wang, D. 2022. D-DPCC: Deep Dynamic Point Cloud Compression via 3D Motion Prediction. In Raedt, L. D., ed., *IJCAI*, 898–904.
- Fu, C.; Li, G.; Song, R.; Gao, W.; and Liu, S. 2022. OctAttention: Octree-Based Large-Scale Contexts Model for Point Cloud Compression. In *AAAI*, 625–633.
- Gao, L.; Fan, T.; Wan, J.; Xu, Y.; Sun, J.; and Ma, Z. 2021. Point Cloud Geometry Compression Via Neural Graph Sampling. In *IEEE Int. Conf. Image Process.*, 3373–3377. IEEE.
- Goyal, V. K. 2001. Theoretical foundations of transform coding. *IEEE Signal Process. Mag.*, 18(5): 9–21.
- He, Y.; Ren, X.; Tang, D.; Zhang, Y.; Xue, X.; and Fu, Y. 2022. Density-preserving Deep Point Cloud Compression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2323–2332. IEEE.
- Hu, Z.; and Xu, D. 2023. Complexity-guided Slimmable Decoder for Efficient Deep Video Compression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 14358–14367.
- Huang, L.; Wang, S.; Wong, K.; Liu, J.; and Urtasun, R. 2020. OctSqueeze: Octree-Structured Entropy Model for LiDAR Compression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1310–1320.
- Huang, T.; and Liu, Y. 2019. 3D Point Cloud Geometry Compression on Deep Learning. In Amsaleg, L.; Huet, B.; Larson, M. A.; Gravier, G.; Hung, H.; Ngo, C.; and Ooi, W. T., eds., *ACM Int. Conf. Multimedia*, 890–898.
- Li, Z.; Wang, W.; Wang, Z.; and Lei, N. 2024. Point Cloud Compression via Constrained Optimal Transport. *CoRR*, abs/2403.08236.
- Liang, Z.; and Liang, F. 2022. TransPCC: Towards Deep Point Cloud Compression via Transformers. In Oria, V.; Sapino, M. L.; Satoh, S.; Kerhervé, B.; Cheng, W.; Ide, I.; and Singh, V. K., eds., *International Conference on Multimedia Retrieval*, 1–5.
- Liu, L.; Hu, Z.; and Zhang, J. 2023. PCHM-Net: A New Point Cloud Compression Framework for Both Human Vision and Machine Vision. In *Int. Conf. Multimedia and Expo*, 1997–2002.
- Mekuria, R.; Blom, K.; and César, P. 2017. Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video. *IEEE Trans. Circuit Syst. Video Technol.*, 27(4): 828–842.
- Minnen, D.; Ballé, J.; and Toderici, G. 2018. Joint Autoregressive and Hierarchical Priors for Learned Image Compression. In *Adv. Neural Inform. Process. Syst.*, 10794–10803.
- Nguyen, D. T.; Quach, M.; Valenzise, G.; and Duhamel, P. 2021. Lossless Coding of Point Cloud Geometry Using a Deep Generative Model. *IEEE Trans. Circuit Syst. Video Technol.*, 31(12).
- Pan, Z.; Xiao, M.; Han, X.; Yu, D.; Zhang, G.; and Liu, Y. 2024. patchDPCC: A Patchwise Deep Compression Framework for Dynamic Point Clouds. In *AAAI*, 4406–4414.
- Qi, Y.; He, Y.; Qi, X.; Zhang, Y.; and Yang, G. 2023. Dynamic Snake Convolution based on Topological Geometric Constraints for Tubular Structure Segmentation. In *Int. Conf. Comput. Vis.*, 6047–6056.
- Quach, M.; Valenzise, G.; and Dufaux, F. 2019. Learning Convolutional Transforms for Lossy Point Cloud Geometry Compression. In *IEEE Int. Conf. Image Process.*, 4320–4324.
- Quach, M.; Valenzise, G.; and Dufaux, F. 2020. Improved Deep Point Cloud Geometry Compression. In *International Workshop on Multimedia Signal Processing*, 1–6.
- Que, Z.; Lu, G.; and Xu, D. 2021. VoxelContext-Net: An Octree Based Framework for Point Cloud Compression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 6042–6051.

- Schwarz, S.; Preda, M.; Baroncini, V.; Budagavi, M.; César, P.; Chou, P. A.; Cohen, R. A.; Krivokuca, M.; Lasserre, S.; Li, Z.; Llach, J.; Mammou, K.; Mekuria, R.; Nakagami, O.; Siahaan, E.; Tabatabai, A. J.; Tourapis, A. M.; and Zakharchenko, V. 2019. Emerging MPEG Standards for Point Cloud Compression. *IEEE J. Emerg. Sel. Topics Circuits Syst.*, 9(1): 133–148.
- Song, R.; Fu, C.; Liu, S.; and Li, G. 2023. Efficient Hierarchical Entropy Model for Learned Point Cloud Compression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 14368–14377.
- Tang, H.; Yang, S.; Liu, Z.; Hong, K.; Yu, Z.; Li, X.; Dai, G.; Wang, Y.; and Han, S. 2023. TorchSparse++: Efficient Training and Inference Framework for Sparse Convolution on GPUs. In *Proceedings of the 56th Annual IEEE/ACM International Symposium on Microarchitecture*, 225–239.
- Tao, L.; Gao, W.; Li, G.; and Zhang, C. 2023. AdaNIC: Towards Practical Neural Image Compression via Dynamic Transform Routing. In *Int. Conf. Comput. Vis.*, 16833–16842.
- Thanou, D.; Chou, P. A.; and Frossard, P. 2016. Graph-Based Compression of Dynamic 3D Point Cloud Sequences. *IEEE Trans. Image Process.*, 25(4): 1765–1778.
- Veit, A.; and Belongie, S. J. 2020. Convolutional Networks with Adaptive Inference Graphs. *Int. J. Comput. Vis.*, 128(3): 730–741.
- Wang, J.; Ding, D.; Chen, H.; and Ma, Z. 2023a. Dynamic Point Cloud Geometry Compression Using Multiscale Inter Conditional Coding. *CoRR*, abs/2301.12165.
- Wang, J.; Ding, D.; Li, Z.; Feng, X.; Cao, C.; and Ma, Z. 2023b. Sparse Tensor-Based Multiscale Representation for Point Cloud Geometry Compression. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(7).
- Wang, J.; Ding, D.; Li, Z.; and Ma, Z. 2021a. Multiscale Point Cloud Geometry Compression. In Bilgin, A.; Marcellin, M. W.; Serra-Sagristà, J.; and Storer, J. A., eds., *Data Compression Conference*, 73–82.
- Wang, J.; Zhu, H.; Liu, H.; and Ma, Z. 2021b. Lossy Point Cloud Geometry Compression via End-to-End Learning. *IEEE Trans. Circuit Syst. Video Technol.*, 31(12): 4909–4923.
- Wang, W.; Dai, J.; Chen, Z.; Huang, Z.; Li, Z.; Zhu, X.; Hu, X.; Lu, T.; Lu, L.; Li, H.; Wang, X.; and Qiao, Y. 2023c. InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 14408–14419.
- Xia, S.; Fan, T.; Xu, Y.; Hwang, J.; and Li, Z. 2023. Learning Dynamic Point Cloud Compression via Hierarchical Interframe Block Matching. In El-Saddik, A.; Mei, T.; Cucchiara, R.; Bertini, M.; Vallejo, D. P. T.; Atrey, P. K.; and Hossain, M. S., eds., *ACM Int. Conf. Multimedia*, 7993–8003.
- Xie, L.; and Gao, W. 2024a. LearningPCC: A PyTorch Library for Learning-Based Point Cloud Compression. In *ACM Int. Conf. Multimedia*.
- Xie, L.; and Gao, W. 2024b. PCHMVision: An Open-Source Library of Point Cloud Compression for Human and Machine Vision. In *ACM Int. Conf. Multimedia*.
- Xie, L.; Gao, W.; Zheng, H.; and Li, G. 2024a. ROI-Guided Point Cloud Geometry Compression Towards Human and Machine Vision. In *ACM Int. Conf. Multimedia*.
- 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*, 594–595.
- Xie, L.; Gao, W.; Zheng, H.; and Ye, H. 2024c. Semantic-Aware Visual Decomposition for Point Cloud Geometry Compression. In *2024 Data Compression Conference (DCC)*, 595–595.
- Xie, L.; Mu, X.; and Gao, W. 2024. PKU-DPCC: A New Dataset for Dynamic Point Cloud Compression. In *APSIPA Transactions on Signal and Information Processing*.
- Yan, W.; Shao, Y.; Liu, S.; Li, T. H.; Li, Z.; and Li, G. 2019. Deep AutoEncoder-based Lossy Geometry Compression for Point Clouds. *CoRR*, abs/1905.03691.
- Yang, C.; Wang, X.; Yao, L.; Long, G.; and Xu, G. 2024. Dyformer: A dynamic transformer-based architecture for multivariate time series classification. *Inf. Sci.*, 656: 119881.
- Yang, F.; Herranz, L.; Cheng, Y.; and Mozerov, M. G. 2021. Slimmable Compressive Autoencoders for Practical Neural Image Compression. In *IEEE Conf. Comput. Vis. Pattern Recog.*
- Zhang, C.; and Gao, W. 2024. Learned Rate Control for Frame-Level Adaptive Neural Video Compression via Dynamic Neural Network. In *Eur. Conf. Comput. Vis.* Springer.
- Zhang, J.; Chen, T.; Ding, D.; and Ma, Z. 2023. YOGA: Yet Another Geometry-based Point Cloud Compressor. In El-Saddik, A.; Mei, T.; Cucchiara, R.; Bertini, M.; Vallejo, D. P. T.; Atrey, P. K.; and Hossain, M. S., eds., *ACM Int. Conf. Multimedia*, 9070–9081. ACM.
- Zuo, D.; Yu, P.; Huang, R.; Huang, Y.; Sun, W.; and Liang, F. 2023. Diverse Context Model for Large-Scale Dynamic Point Cloud Compression. In *VCIP*, 1–5.