D-DPCC: Deep Dynamic Point Cloud Compression via 3D Motion Prediction

Tingyu Fan[*1] , **Linyao Gao**[*1] , **Yiling Xu**[1] , **Zhu Li**[2] and **Dong Wang**[3]
[1]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University
[2]University of Missouri, Kansas City
[3]Guangdong OPPO Mobile Telecommunications Corp., Ltd.
{woshiyizhishapaozi, linyaog, yl.xu}@sjtu.edu.cn, zhu.li@ieee.org, wangdong7@oppo.com

Abstract

The non-uniformly distributed nature of the 3D dynamic point cloud (DPC) brings significant challenges to its high-efficient inter-frame compression. This paper proposes a novel 3D sparse convolution-based Deep Dynamic Point Cloud Compression (D-DPCC) network to compensate and compress the DPC geometry with 3D motion estimation and motion compensation in the feature space. In the proposed D-DPCC network, we design a *Multi-scale Motion Fusion* (MMF) module to accurately estimate the 3D optical flow between the feature representations of adjacent point cloud frames. Specifically, we utilize a 3D sparse convolution-based encoder to obtain the latent representation for motion estimation in the feature space and introduce the proposed MMF module for fused 3D motion embedding. Besides, for motion compensation, we propose a 3D *Adaptively Weighted Interpolation* (3DAWI) algorithm with a penalty coefficient to adaptively decrease the impact of distant neighbors. We compress the motion embedding and the residual with a lossy autoencoder-based network. To our knowledge, this paper is the first work proposing an end-to-end deep dynamic point cloud compression framework. The experimental result shows that the proposed D-DPCC framework achieves an average 76% BD-Rate (Bjontegaard Delta Rate) gains against state-of-the-art Video-based Point Cloud Compression (V-PCC) v13 in inter mode.

1 Introduction

In recent years, the dynamic point cloud (DPC) has become a promising data format for representing sequences of 3D objects and scenes, with broad applications in AR/VR, autonomous driving, and robotic sensing [Zhang *et al.*, 2021]. However, compared with pixelized 2D image/video, the non-uniform distribution of DPC makes the exploration of temporal correlation extremely difficult, bringing significant challenges to its inter-frame compression. This paper focuses on the dynamic point cloud geometry compression with 3D motion estimation and motion compensation to reduce temporal redundancies of 3D point cloud sequences.

The existing dynamic point cloud compression (DPCC) methods can be concluded as 2D-video-based and 3D-model-based methods [Li *et al.*, 2021]. For 2D-video-based DPCC, the Moving Picture Expert Group (MPEG) proposes Video-based Point Cloud Compression (V-PCC) [Schwarz *et al.*, 2018], which projects DPC into 2D geometry and texture video and uses mature video codecs (e.g., HEVC) for high-efficient video compression. Among all DPCC methods, V-PCC achieves current state-of-the-art performance. On the other hand, the 3D-model-based methods rely on motion estimation and motion compensation on 3D volumetric models like octree [de Queiroz and Chou, 2017]. However, the above methods are rule-based, with hand-crafted feature extraction modules and assumption-based matching rules, resulting in unsatisfactory coding efficiency.

Recently, the end-to-end learnt static point cloud (SPC) compression methods have reported remarkable gains against traditional SPC codecs like Geometry-based PCC (G-PCC) and V-PCC (intra) [Xu *et al.*, 2018]. Most of the learnt SPC compression methods are built upon the autoencoder architecture for dense object point clouds [Wang *et al.*, 2021d; Gao *et al.*, 2021; Wang *et al.*, 2021c], which divide SPC compression into three consecutive steps: feature extraction, deep entropy coding, and point cloud reconstruction. However, it is non-trivial to migrate the SPC compression networks to DPC directly. The critical challenge is to embed the motion estimation and motion compensation into the end-to-end compression network to remove temporal redundancies.

To this end, we propose a Deep Dynamic Point Cloud Compression (D-DPCC) framework, which optimizes the motion estimation, motion compensation, motion compression, and residual compression module in an end-to-end manner. Our contributions are summarized as follows:

1. We first propose an end-to-end Deep Dynamic Point Cloud Compression framework (D-DPCC) for the joint optimization of motion estimation, motion compensation, motion compression, and residual compression.
2. We propose a novel Multi-scale Motion Fusion (MMF) module for point cloud inter-frame prediction, which extracts and fuses the motion flow information at different

[*]Equal contribution.

scales for accurate motion estimation.

3. For motion compensation, we propose a novel 3D Adaptively Weighted Interpolation (3DAWI) algorithm, which utilizes the neighbor information and adaptively decreases the impact of distant neighbors to produce a point-wise prediction of the current frame's feature.
4. Experimental result shows that our framework outperforms the state-of-the-art V-PCC v13 by an average 76% Bjontegaard Delta Rate (BD-Rate) gain on the 8iVFB [d'Eon *et al.*, 2017] dataset suggested by MPEG.

2 Related Work

Dynamic Point Cloud Compression. Existing DPCC works can be mainly summarized as 2D-video-based and 3D-model-based methods. The 2D-video-based methods perform 3D-to-2D projection to utilize the 2D motion estimation (ME) and motion compensation (MC) algorithms. Among them, MPEG V-PCC [Schwarz *et al.*, 2018] reports state-of-the-art performance. On the contrary, the 3D-model-based methods directly process the original 3D DPC. [Thanou *et al.*, 2015] adopts geometry-based graph matching for attribute coding. [de Queiroz and Chou, 2017] breaks the DPC into blocks of voxels and performs block-based matching for 3D ME and MC to encode the point cloud geometry and attribute. However, the 3D-model-based methods fail to fully exploit the temporal correlations, leading to inferior results than 2D-video-based methods.

Learnt Static Point Cloud Compression. Inspired by the implementation of deep learning techniques in image/video compression, recently learnt static point cloud compression (SPCC) networks have emerged. [Huang and Liu, 2019; Gao *et al.*, 2021] design point-based SPCC networks to compress the 3D point clouds directly. However, these works are limited to small-scale point cloud datasets with only thousands of points. Wang et al. [Wang *et al.*, 2021d] utilize 3D CNN-based frameworks for voxelized point cloud compression, achieving remarkable compression efficiency at the cost of excessive time and memory complexity. Thanks to the development of 3D sparse convolution [Choy *et al.*, 2019], Wang et al. [Wang *et al.*, 2021c] propose an end-to-end sparse convolution-based multi-scale SPCC network, which reports state-of-the-art performance on large-scale datasets.

Point Cloud Scene Flow Estimation. Recently, 3D scene flow estimation has received much attention, which inspires us to compress DPC geometry by 3D motion estimation. FlowNet3D [Liu *et al.*, 2019] is the representative work of 3D scene flow estimation, which integrates a set conv layer based on PointNet++ [Qi *et al.*, 2017] for feature extraction, followed by a flow embedding layer for point matching. HALFlow [Wang *et al.*, 2021b] proposes a double-attentive flow-embedding layer, which learns the correlation weights from both relative coordinates and feature space. Res3DSF [Wang *et al.*, 2021a] proposes a context-aware set conv layer and a residual flow learning structure for repetitive pattern recognition and long-distance motion estimation, which reports state-of-the-art performance. However, the above methods are trained on manually labelled scene flow data, leading to inefficient motion estimation for compression tasks. Furthermore, the above methods suffer from excessive time complexity, which are inapplicable on large-scale DPCs with hundreds of thousands of points in each frame.

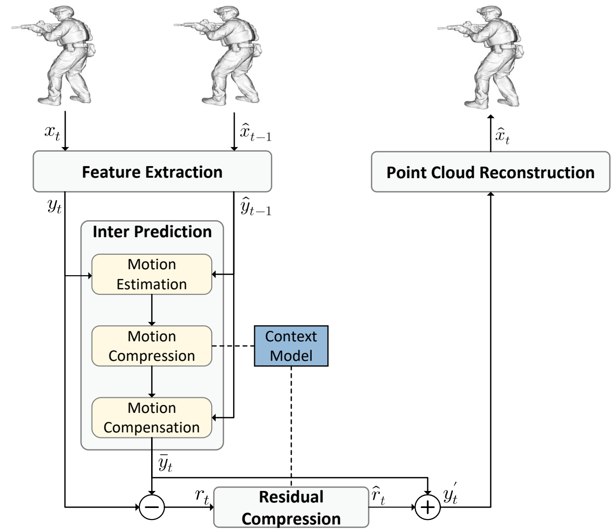

Figure 1: The overall architecture of D-DPCC. x_t and $\hat{x}_{t-1}$ are the current frame and the previously reconstructed frame. y_t and $\hat{y}_{t-1}$ are the associated latent representations in feature space. $\bar{y}_t$ is the prediction of y_t. r_t and $\hat{r}_t$ are the feature residual and reconstructed residual. y_t' is the reconstruction value of y_t in decoder.

Learnt Video Compression. There are massive attempts to apply deep learning techniques to video compression and build an end-to-end framework. Among them, DVC [Lu *et al.*, 2019] is the first video compression deep model that jointly optimizes all the components for video compression. For more accurate motion estimation, [Hu *et al.*, 2021] further proposes FVC by performing inter-frame prediction in the feature space, reporting superior performance over DVC.

3 Methodology

3.1 Overview

This section introduces the proposed end-to-end dynamic point cloud compression framework D-DPCC. The overall architecture of D-DPCC is shown in Figure 1. Let $x_t = \{C_{x_t}, F_{x_t}\}$ and $x_{t-1} = \{C_{x_{t-1}}, F_{x_{t-1}}\}$ be two adjacent point cloud frames, where C_{x_t} and $C_{x_{t-1}}$ are coordinate matrices. F_{x_t} and $F_{x_{t-1}}$ are the associated feature matrices with all-one vectors to indicate voxel occupancy. The network analyses the correlation between x_t and the previously reconstructed frame $\hat{x}_{t-1}$, aiming to reduce the bit rate consumption with inter-frame prediction. Specifically, x_t and $\hat{x}_{t-1}$ are first encoded into latent representation y_t and $\hat{y}_{t-1}$ in the feature extraction module. Then y_t and $\hat{y}_{t-1}$ are fed into the inter prediction module to generate the predicted latent representation $\bar{y}_t$ of the current frame. The residual compression module compresses the feature residual r_t between y_t and $\bar{y}_t$. Finally, the reconstructed residual $\hat{r}_t$ is summed up with $\hat{y}_t$ and passes through the point cloud reconstruction module to produce the reconstruction $\hat{x}_t$ of the current frame.

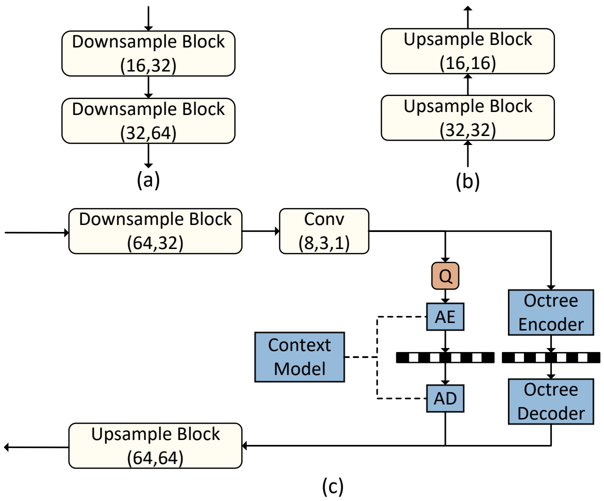

Figure 2: The network architecture of (a) Feature Extraction, (b) Point Cloud Reconstruction, (c) Residual Compression.

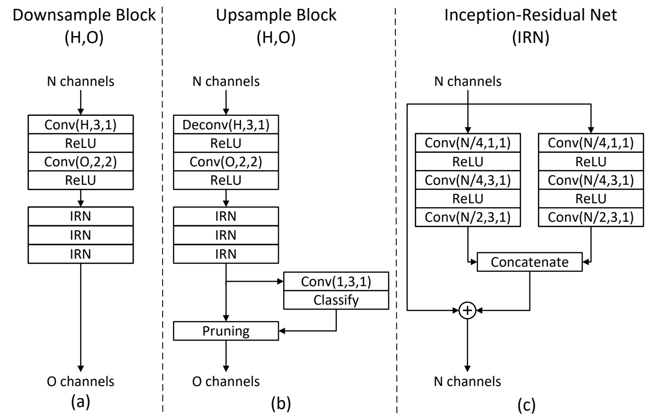

Figure 3: The detailed architecture of each block. (a) Downsample Block, (b) Upsample Block, (c) Inception-Residual Block Net (IRN) block. H and O indicates the number of hidden and output channels.

3.2 Feature Extraction

The feature extraction module (Figure 2(a)) consists of two serially connected Downsample Blocks for the hierarchical reduction of spatial redundancies, which encodes the current frame x_t and the previously reconstructed frame $\hat{x}_{t-1}$ as latent representation y_t and $\hat{y}_{t-1}$. Inspired by [Wang *et al.*, 2021c], we adopt the sparse CNN-based Downsample Block (Figure 3(a)) for low-complexity point cloud downsampling. The Downsample Block consists of a stride-two sparse convolution layer for point cloud downsampling, followed by several Inception-Residual Network (IRN) blocks [Szegedy *et al.*, 2017] (Figure 3(c)) for local feature analysis and aggregation. For simplicity of notations, we denote the coordinate matrix of x_t at different downsampling scales as $C^k_{x_t}$, where k is the scale index (i.e., $\times\frac{1}{2^k}$). Correspondingly, $y_t = \{C^2_{x_t}, F_{y_t}\}$, where F_{y_t} is the feature matrix of y_t.

3.3 Inter Prediction

The inter prediction module takes the latent representation of both the current frame and the previously reconstructed frame, i.e., y_t and $\hat{y}_{t-1}$ as input, analysing the temporal correlation and producing the feature prediction of y_t, i.e., $\bar{y}_t$.

The existing 3D scene flow estimation networks estimate a motion flow between two frames p_1 and p_2: $\mathcal{D} = \{x_i, y_i, z_i, \mathbf{d}_i | i = 1, \cdots, N_1\}$ to minimize the distortion between $\mathcal{D}$ and the ground truth $\mathcal{D}^*$ [Liu *et al.*, 2019]. However, without end-to-end optimization, the predicted motion flow is unsatisfactory for motion compensation and motion compression. Therefore, we borrow the idea of feature-space motion compensation [Hu *et al.*, 2021] from video compression to design the end-to-end inter prediction module.

The overall architecture of the inter prediction module is shown in Figure 4. Specifically, we first *concatenate* y_t and $\hat{y}_{t-1}$ together to get y^{cat}_t. The *concatenate* operation for point clouds is defined as:

$$x^{cat}_u = \begin{cases} x_{1,u} \oplus x_{2,u} & u \in C_1 \cap C_2 \\ \mathbf{0} \oplus x_{2,u} & u \notin C_1 \wedge u \in C_2 \\ x_{1,u} \oplus \mathbf{0} & u \in C_1 \wedge u \notin C_2 \end{cases} \quad (1)$$

where x^{cat}_u is the concatenation of x_1 and x_2. $x_{1,u}$ and $x_{2,u}$ are the input feature vectors defined at u, and $\oplus$ is the concatenate operation between vectors. y^{cat}_t passes through a 2-layer convolution network to generate the original flow embedding $e_{o,t}$. We further design a Multi-scale Motion Fusion (MMF) module for precise motion flow estimation. MMF extracts multi-scale flow embedding from $e_{o,t}$ and outputs a fused flow embedding e_t, which is encoded by an autoencoder-style network (Figure 4(b)), whose encoder and decoder are each composed of a stride-two sparse convolution and transpose convolution layer, respectively.

Multi-scale Motion Fusion. Due to the sparsity nature of point clouds, the Euclidean distances between two adjacent frames' corresponding points have an enormous variance, leading to ineffective matching between points in y_t and $\hat{y}_{t-1}$. A straightforward solution is to increase the downsampling factor to expand the perception field. However, this leads to more information loss. Therefore, We propose a Multi-scale Motion Fusion (MMF) module as shown in Figure 5. The MMF module enhances the original flow embedding $e_{o,t}$ with the multi-scale flow information, which first downsamples $e_{o,t}$ with a stride-two sparse convolution layer to expand the perception field, then uses several Residual Net (RN) blocks to generate the coarse-grained flow embedding $e_{c,t}$. To compensate for the information loss during downsampling, we compute the residual (denoted as Δe_t) between upsampled $e_{c,t}$ and $e_{o,t}$, then downsample Δe_t to obtain the fine-grained flow embedding $e_{f,t}$. Finally, $e_{c,t}$ is summed up with $e_{f,t}$ to generate the fused flow embedding e_t. Additionally, for multi-scale motion compensation, we design a Multi-scale Motion Reconstruction (MMR) module, which is symmetric to MMF. MMR recovers coarse-grained and fine-grained motion flow $m_{c,t}$ and $m_{f,t}$ from the reconstructed flow embedding $\hat{e}_t$, where $m_{c,t}$ is up-scaled and summed up with $m_{f,t}$ to generate the fused motion flow m_t.

3D Adaptively Weighted Interpolation. The neural network estimates a continuous motion flow m_t, without the explicit specification to the corresponding geometry coordinates in the previous frame. Therefore, we propose a 3D Adap-

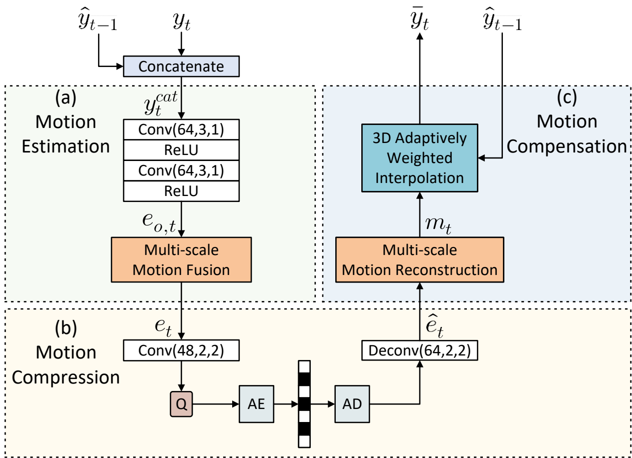

Figure 4: The network architecture of the inter prediction module. (a) Motion estimation, (b) Motion compression, (c) Motion compensation. $e_{o,t}$ and e_t are the original flow embedding and fused flow embedding. $\hat{e}_t$ is the reconstructed flow embedding. m_t is the estimated motion flow.

tively Weighted Interpolation (3DAWI) algorithm to obtain the derivable predicted latent representation $\bar{y}_t$ of the current frame. Given m_t and $\hat{y}_{t-1}$, the predicted latent representation $\bar{y}_t$ is derived as follows:

$$\bar{y}_{t,u} = \frac{\sum_{v \in \vartheta_{u'}} d_{u',v}^{-1} \cdot \hat{y}_{t-1,v}}{\max(\sum_{v \in \vartheta_{u'}} d_{u',v}^{-1}, \alpha)} \text{ for } u \text{ in } \mathrm{C}^2_{\mathrm{x_t}}, \tag{2}$$

where $\bar{y}_{t,u}$ is the predicted feature vector of $\bar{y}_t$ defined at u, $\hat{y}_{t-1,v}$ is the feature vector of $\hat{y}_{t-1}$ defined at v, u' is u's translated coordinate, $\vartheta_{u'}$ is the 3-nearest neighbors of u' in $\hat{y}_{t-1}$, and $d_{u',v}$ is the squared Euclidean distance between coordinates u' and v. α is a penalty coefficient related to point-spacing. Here, we intuitively give its value as $\alpha = 3m$, where m is the minimum point spacing in the dataset. The 3DAWI uses inverse distance weighted average (IDWA) based on geometric 3-nearest neighbors [Qi *et al.*, 2017], which has a broader perception field than the trilinear interpolation algorithm. However, the IDWA based interpolation algorithm assigns an equal weighted sum of 1 for each u in $C^2_{x_t}$, inappropriate for u with less effective corresponding points in $\hat{y}_{t-1}$. Therefore, 3DAWI utilizes α to adaptively decrease the weighted sum of each u in $C^2_{x_t}$ whose translated coordinate u' is *isolated* in $\hat{y}_{t-1}$, where u' is *isolated* indicates that $\sum_{i \in \vartheta_{u'}} d_{u',i}^{-1} < \alpha$, i.e., u' deviates from its 3-nearest neighbors, and the matching is less effective. Under extreme conditions, given $d_{u',v} \to \infty$ for each v in $\vartheta_{u'}$, then $\bar{y}_{t,u} \to \mathbf{0}$. Note that the coordinate matrix of y_t, i.e., $C^2_{x_t}$, is required in Equation (2), which is losslessly encoded [Nguyen *et al.*, 2021] and transmitted to the decoder, and only accounts for a small amount (< 0.05 bpp) of the whole bitstream. (See supplementary material for more details).

3.4 Residual Compression

The residual compression module encodes the residual r_t between y_t and $\bar{y}_t$ in feature space. We employ an autoencoder-style lossy compression network for residual compression as shown in Figure 2(c). The encoder consists of a Downsample Block and a sparse convolutional layer, aiming to transform r_t into latent residual $l_t = \{C^3_{x_t}, F_{l_t}\}$. $C^3_{x_t}$ is losslessly coded using G-PCC (octree) [Schwarz *et al.*, 2018], and the feature matrix F_{l_t} is quantized and coded with a factorized entropy model [Ballé *et al.*, 2017]. The decoder uses an Upsample Block (Figure 3(b)) to recover the reconstructed residual $\hat{r}_t$, which is added with the predicted latent representation $\bar{y}_t$ to obtain the reconstructed feature y'_t as shown in Figure 1.

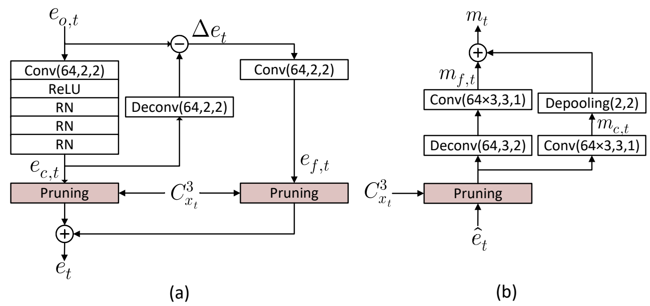

Figure 5: The architecture of (a) Multi-scale Motion Fusion (MMF) module, (b) Multi-scale Motion Reconstruction (MMR) module.

3.5 Point Cloud Reconstruction

The point cloud reconstruction module (Figure 2(b)) mirrors the operations of feature extraction, with two serially connected Upsample Block (Figure 3(b)) for hierarchical reconstruction. The Upsample Block uses a stride-two sparse transpose convolution layer for point cloud upsampling. After successive convolutions, the Upsample Block uses a sparse convolution layer to generate the occupation probability of each voxel. To maintain the sparsity of the point cloud after upsampling, we employ adaptive pruning [Wang *et al.*, 2021c] to detach false voxels based on the occupation probability.

3.6 Loss Function

We apply the rate-distortion joint loss function for end-to-end optimization:

$$\mathcal{L} = \mathcal{R} + \lambda \mathcal{D}, \tag{3}$$

where $\mathcal{R}$ is the bits per point (bpp) for encoding the current frame x_t, and $\mathcal{D}$ is the distortion between x_t and the decoded current frame $\hat{x}_t$.

Rate. The latent feature f is quantized before encoding. Note that the quantization operation is non-differentiable, thus we approximate the quantization process by adding a uniform noise $\mu \sim \mathcal{U}(-0.5, 0.5)$. The quantized latent feature $\bar{f}$ is encoded using arithmetic coding with a factorized entropy model [Ballé *et al.*, 2017], which estimates the probability distribution of $\bar{f}$, i.e., $p_{\tilde{f}|\psi}(\tilde{f}|\psi)$, where ψ are the learnable parameters. Then the bpp of encoding $\tilde{f}$ is:

$$\mathcal{R} = \frac{1}{N} \sum_i \log_2 p_{\tilde{f}_i|\psi^{(i)}}(\tilde{f}_i|\psi^{(i)}), \tag{4}$$

where N is the number of points in x_t, and i is the index of channels.

Distortion. As shown in Figure 3(b), in the proposed Downsample Block, a sparse convolution layer is used to produce the occupation probability p_v of each voxel v in the decoded point cloud. Therefore, we apply binary cross entropy (BCE) loss to measure the distortion:

$$\mathcal{D}_{BCE} = \frac{1}{N}\sum_{v} -(\mathcal{O}_v \log p_v + (1-\mathcal{O}_v)\log(1-p_v)), \quad (5)$$

where $\mathcal{O}_v$ is the ground truth that either v is occupied (1) or unoccupied (0). For hierarchical reconstruction, the distortion of each scale is averaged, i.e.,

$$\mathcal{D} = \frac{1}{K}\sum_{k=1}^{K}\mathcal{D}_{BCE}^{(k)}, \quad (6)$$

where k is the scale index.

4 Experiments

4.1 Experimental Settings

Training Dataset. We train the proposed model using Owlii Dynamic Human DPC dataset [Keming *et al.*, 2018], containing 4 sequences with 2400 frames. The frame rate is 30 frames per second (fps) over 20 seconds for each sequence. To reduce the time and memory consumption during training and exemplify the scalability of our model, we quantize the 11-bit precision point cloud data into 9-bit precision.

Evaluating Dataset. Following the MPEG common test condition (CTC), we evaluate the performance of the proposed D-DPCC framework using 8i Voxelized Full Bodies (8iVFB) [d'Eon *et al.*, 2017], containing 4 sequences with 1200 frames. The frame rate is 30 fps over 10 seconds.

Training Strategy. We train D-DPCC with λ =3, 4, 5, 7, 10 for each rate point. We utilize an Adam [Kingma and Ba, 2015] optimizer with $\beta = (0.9, 0.999)$, together with a learning rate scheduler with a decay rate of 0.7 for every 15 epochs. A two-stage training strategy is applied for each rate point. Specifically, for the first five epochs, λ is set as 20 to accelerate the convergence of the point cloud reconstruction module; then, the model is trained for another 45 epochs with λ set to its original value. The batch size is 4 during training. We conduct all the experiments on a GeForce RTX 3090 GPU with 24GB memory.

Evaluation Metric. The bit rate is evaluated using bits per point (bpp), and the distortion is evaluated using point-to-point geometry (D1) Peak Signal-to-Noise Ratio (PSNR), and point-to-plane geometry (D2) PSNR following the MPEG CTC. The peak value p is set as 1023 for 8iVFB.

4.2 Experimental Results

Baseline Setup. We compare the proposed D-DPCC with the current state-of-the-art dynamic point cloud geometry compression framework: V-PCC Test Model v13, with the quantization parameter (QP) setting as $18, 15, 12, 10, 8$, respectively. We also compare with Wang's framework [Wang *et al.*, 2021c], which is state-of-the-art on static object point

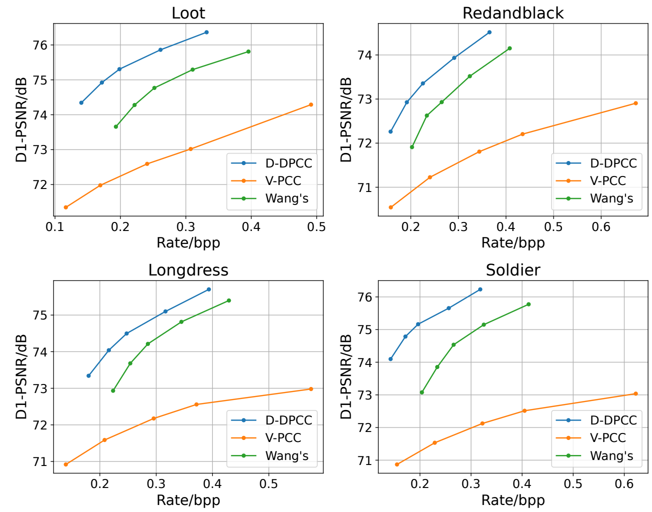

Figure 6: D1 Rate-Distortion curves on 8iVFB (Loot, Redandblack, Longdress, Soldier) test sequences.

Sequence	V-PCC		Wang's	
	D1	D2	D1	D2
Loot	-71.76	-70.92	-36.15	-29.73
Redandblack	-68.97	-76.71	-25.68	-20.56
Soldier	-85.71	-73.08	-39.48	-36.47
Longdress	-78.92	-72.89	-22.31	-17.91
Overall	**-76.66**	**-74.43**	**-31.14**	**-26.39**

Table 1: BD-Rate(%) gains against V-PCC (inter) and Wang's framework. For the proposed D-DPCC, the first frame of a sequence is encoded by Wang's network. The encoding of each subsequent frame takes the previous reconstructed frame as a reference.

cloud geometry compression. For the fairness of comparison, we retrain Wang's framework using our training data and strategy, and the network parameters of each module are set the same as Wang's except for the proposed inter prediction module. When using the proposed D-DPCC for inter-frame coding, the first frame of the sequence is encoded using Wang's network with the same λ.

Performance Analysis. The rate-distortion curves of different methods are presented in Figure 6, and the corresponding BD-Rate gains are shown in Table 1. Our D-DPCC outperforms V-PCC v13 by a large margin on all test sequences, with an average 76.66% (D1), 74.43% (D2) BD-Rate gain. Compared with Wang's network, the experimental result demonstrates that D-DPCC significantly improves the coding efficiency on test sequences Loot and Soldier with small motion amplitude, achieving $> 36\%$ (D1), $> 29\%$ (D2) BD-Rate gains. Meanwhile, D-DPCC still reports a considerable bit-rate reduction on Longdress and Redandblack with larger motion amplitude with $> 22\%$ (D1), $> 17\%$ (D2) BD-Rate gain. On all test sequences, D-DPCC achieves an average 31.14% (D1) and 26.39% (D2) BD-Rate gain against Wang's network. D-DPCC integrates a learnt inter prediction module with an MMF module for accurate motion estima-

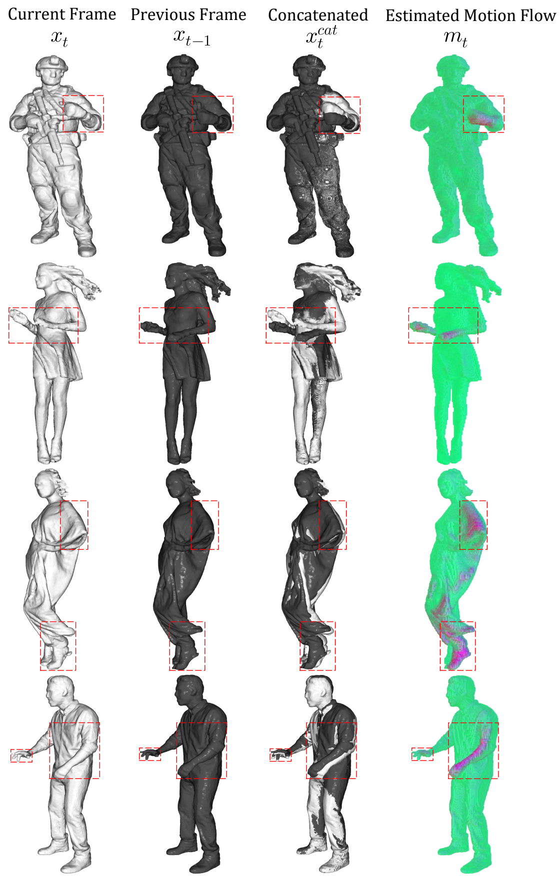

Figure 7: Visualization of the estimated motion flow. The leftmost column is the current frames; the left middle column is the previous frames; the middle right column is the concatenations of the current frames and the previous frames. The rightmost column visualizes the predicted motion flow, with green indicating stillness and purple indicating movement.

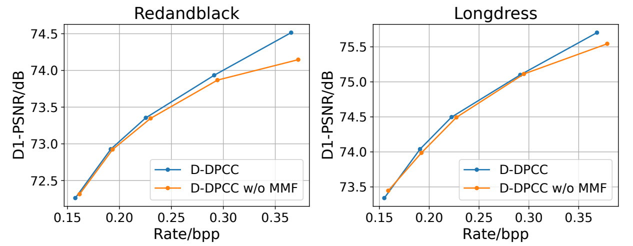

Figure 8: Ablation study on Redandblack and Longdress for Multi-scale Motion Fusion (MMF) module.

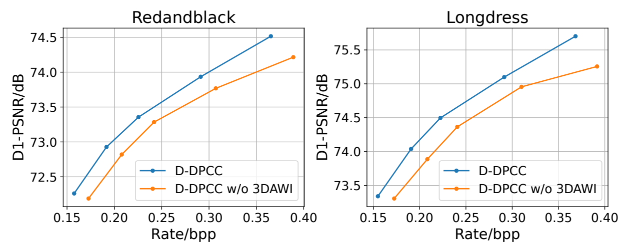

Figure 9: Ablation study on Redandblack and Longdress for 3D Adaptively Weighted Interpolation (3DAWI).

tion and the 3DAWI algorithm for point-wise motion compensation, which is jointly optimized with the other modules in D-DPCC. Therefore, D-DPCC significantly surpasses V-PCC (inter) and Wang's method.

4.3 Analysis and Ablation Study

Effectiveness of Inter Prediction. To verify the effectiveness of the inter prediction module, we visualize the estimated motion flow m_t between two adjacent frames in Figure 7. It can be observed that although no ground truth of 3D motion flow is provided during training, the inter prediction module still learns an explainable motion flow based on the analysis of movements between two adjacent point cloud frames through end-to-end training.

Effectiveness of Multi-scale Motion-fusion. Figure 8 reports the D1 rate-distortion curves of D-DPCC with/without the Multi-scale Motion Fusion (MMF) module. It is noted that with the multi-scale flow embedding and the fusion of the coarse-grained and fine-grained motion flow, MMF improves the coding efficiency by $> 4\%$ BD-Rate gains. The improvement is particularly significant at high bit rates, where the network is encouraged to produce precise motion flow estimation. The performance improvement clearly demonstrates the effectiveness of the proposed MMF module.

Effectiveness of 3D Adaptively Weighted Interpolation. The objective performance with/without the penalty coefficient α of 3D Adaptively Weighted Interpolation (3DAWI) is presented in Figure 9. It can be observed that 3DAWI significantly improves the objective performance, with 12.63% D1 BD-Rate gain on Redandblack and 14.33% D1 BD-Rate gain on Longdress. 3DAWI produces a point-wise feature prediction and penalizes those *isolated* points that deviate from their 3-nearest neighbors in the referenced point cloud, by adaptively decreasing their weighted sum. Therefore, 3DAWI brings significant improvement in coding efficiency. The minimum distance between points in 8iVFB is 1; thus, α is set as 3 in experiments.

5 Conclusion

This paper proposes a Deep Dynamic Point Cloud Compression (D-DPCC) framework for the compression of dynamic point cloud geometry sequence. We introduce an inter prediction module to reduce the temporal redundancies. We also propose a Multi-scale Motion Fusion (MMF) module to extract the multi-scale motion flow. For motion compensation, a 3D Adaptively Weighted Interpolation (3DAWI) algorithm is introduced. The proposed D-DPCC achieves 76.66% BD-Rate gain against state-of-the-art V-PCC Test Model v13.

Acknowledgements

This paper is supported in part by National Key R&D Program of China (2018YFE0206700), National Natural Science Foundation of China (61971282, U20A20185). The corresponding author is Yiling Xu (e-mail: yl.xu@sjtu.edu.cn).

References

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

[Choy *et al.*, 2019] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[de Queiroz and Chou, 2017] Ricardo L de Queiroz and Philip A Chou. Motion-compensated compression of dynamic voxelized point clouds. *IEEE Transactions on Image Processing*, 26(8):3886–3895, 2017.

[d'Eon *et al.*, 2017] Eugene d'Eon, Bob Harrison, Taos Myers, and Philip A Chou. 8i voxelized full bodies-a voxelized point cloud dataset. *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006*, 7:8, 2017.

[Gao *et al.*, 2021] Linyao Gao, Tingyu Fan, Jianqiang Wang, Yiling Xu, Jun Sun, and Zhan Ma. Point cloud geometry compression via neural graph sampling. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3373–3377. IEEE, 2021.

[Hu *et al.*, 2021] Zhihao Hu, Guo Lu, and Dong Xu. Fvc: A new framework towards deep video compression in feature space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1502–1511, 2021.

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

[Keming *et al.*, 2018] Cao Keming, Xu Yi, Lu Yao, and Wen Ziyu. Owlii dynamic human mesh sequence dataset. *Document ISO/IEC JTC1/SC29/WG11 m42816, San Diego*, 2018.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations, ICLR 2015*, 2015.

[Li *et al.*, 2021] Zhu Li, Shan Liu, Frederic Dufaux, Li Li, Ge Li, and C.-C. Jay Kuo. Guest editorial introduction to the special issue on recent advances in point cloud processing and compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(12), 2021.

[Liu *et al.*, 2019] Xingyu Liu, Charles R Qi, and Leonidas J Guibas. Flownet3d: Learning scene flow in 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 529–537, 2019.

[Lu *et al.*, 2019] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11006–11015, 2019.

[Nguyen *et al.*, 2021] Dat Thanh Nguyen, Maurice Quach, Giuseppe Valenzise, and Pierre Duhamel. Learning-based lossless compression of 3d point cloud geometry. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4220–4224. IEEE, 2021.

[Qi *et al.*, 2017] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5105–5114, 2017.

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

[Szegedy *et al.*, 2017] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[Thanou *et al.*, 2015] Dorina Thanou, Philip A Chou, and Pascal Frossard. Graph-based motion estimation and compensation for dynamic 3d point cloud compression. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3235–3239. IEEE, 2015.

[Wang *et al.*, 2021a] Guangming Wang, Yunzhe Hu, Xinrui Wu, and Hesheng Wang. Residual 3d scene flow learning with context-aware feature extraction. *arXiv preprint arXiv:2109.04685*, 2021.

[Wang *et al.*, 2021b] Guangming Wang, Xinrui Wu, Zhe Liu, and Hesheng Wang. Hierarchical attention learning of scene flow in 3d point clouds. *IEEE Transactions on Image Processing*, 30:5168–5181, 2021.

[Wang *et al.*, 2021c] Jianqiang Wang, Dandan Ding, Zhu Li, and Zhan Ma. Multiscale point cloud geometry compression. In *2021 Data Compression Conference (DCC)*, pages 73–82. IEEE, 2021.

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

[Xu *et al.*, 2018] Yiling Xu, Ke Zhang, Lanyi He, Zhiqian Jiang, and Wenjie Zhu. Introduction to point cloud compression. *ZTE Communications*, 16(3):8, 2018.

[Zhang *et al.*, 2021] Gege Zhang, Qinghua Ma, Licheng Jiao, Fang Liu, and Qigong Sun. Attan: Attention adversarial networks for 3d point cloud semantic segmentation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 789–796, 2021.