SUPPLEMENTARY MATERIAL FOR "UH-PCC: UNIFIED OCTREE AND FEATURE CODING FOR HIERARCHICAL POINT CLOUD GEOMETRY COMPRESSION"

1. INTRODUCTION

In this supplementary material, we provide more details stochastic training strategy used for training the proposed unified model, and some additional details about the datasets used for experiments.

2. LEVELWISE STOCHASTIC TRAINING

Since our proposed model is unified for both Octree Coding and Feature Coding, a special mechanism is applied during training. As pointed out in Sec. 2 in the main text, our method breaks the dependency between levels in terms of feature extraction (while maintaining the dependency in terms of coding). A motivation for this design is to enable the training to be performed in a level-wise manner rather than fully end-to-end, so as to reduce computational cost. Additionally, since our model is unified between Octree Coding and Feature Coding in terms of network parameters (majority of the parameters) and can seamlessly switch between inter and intra modes with the proposed Style Control technique, we can randomly switch between four coding configurations during training: intra Octree, intra Feature, inter Octree, and inter Feature. The ratio of our network getting trained in either configuration can be controlled by user-specified parameters.

To avoid our network getting stuck in local minimas that lead to poor inference performance, we employ a two-stage training procedure with a special schedule for rate-distortion trade-off parameter λ. In the first stage, we use a fixed low value λ which incurs a high bitrate during training but can achieve good reconstruction performance, *i.e.*, low distortion. This first stage lasts a few epochs, after which we use a normal schedule for λ where we randomly pick the value for λ within a specified range.

For the training of Octree coding the training loss consists of a binary cross-entropy loss (between the predicted probabilities and the ground truth occupancies) used for learning the occupancy probabilities and the rate loss of the features which is the bitrate estimated by the entropy bottleneck layer. The Feature Coding is trained using a rate distortion loss $L = L_D + \lambda L_R$, where the rate is again the bitrate of the feature estimated by the entropy bottleneck layer, and the distortion is computed using the binary cross-entropy loss as the Feature Coding produces a lossy utilizing the learned occupancy probabilities.

Table 1: Training datasets used.

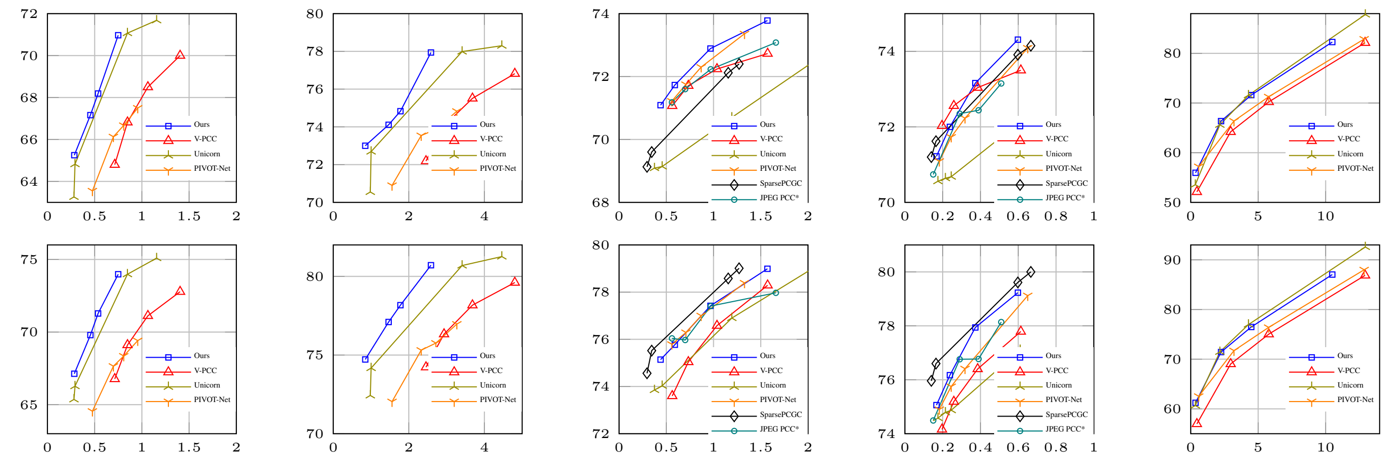
Class	Train (Sequence) Name	Fr.	Prc.
DS	Head_00039_vox12	1	12
	Frog_00067_vox12	1	12
	Egyptian_mask_vox12	1	12
	ULB_Unicorn_vox13	1	13
DS	RWTT Train Set	406	Float
DD	Queen	250	10
	8i VFB – Loot	300	10
	8i VFB – Red_and_Black	300	10
	8i VFB – Soldier	300	10
	8i VFB – Long_dress	300	10
SD	KITTI (00-10)	23201	18

Table 2: Test datasets used.

Class	Test (Sequence) Name	Fr.	Prc.
DS	Facade_00009_vox12	1	12
	House_without_roof_00057_vox12	1	12
	Arco_Valentino_Dense_vox12	1	12
	Statue_Klimt_vox12	1	12
	Shiva_00035_vox12	1	12
DS	RWTT_vishnu_156_vox10	1	10
	RWTT_foxstatue_211_vox10	1	10
	RWTT_tomb_059_vox10	1	10
DD	Exercise_vox10	300	10
	Model_vox10	300	10
	Dancer_vox11	300	11
	Basketball_player_vox11	300	11
	Thaidancer_viewdep_vox12	300	12
SD	KITTI (11-21)	3300	18

3. DATASET DETAILS

The datasets for our experimental evaluation consist of three main categories: Dense Static, Dense Dynamic and Sparse Dynamic, and follows the experimental guidelines outlined in the MPEG AI-PCC CfP [1]. The training set in the Dense Static category consists of a subset of the static point clouds from MPEG G-PCC CTC and some subset of static 3D tex-



(a) "exercise_vox10" (b) "dancer_vox11" (c) "facade_00009_vox12" (d) "house" (e) "KITTI11"

Fig. 1: R-D performance on selected point clouds measured in D1- (top row) and D2- (bottom row) PSNRs. UH-PCC exhibits competitive performance among all the methods. Note that the name "house_without_roof_00057_vox12" is abbreviated to "house" for brevity.

tured models in the Real World Textured Things (RWTT) dataset [2] with permittive licenses. The corresponding test sets consist of another subset of the static point clouds from MPEG G-PCC CTC and three static 3D textured models in the RWTT dataset [1].

In the Dense Dynamic category, the training set is composed of 8iVFB dynamic point cloud sequences and the Queen sequence, while the test set consists of dynamic point cloud sequences from the V-PCC CTC. Finally, the training and testing data for Sparse Dynamic are the training and testing splits from the well-known spinning-LiDAR KITTI [3] dataset. Some additional details about each dataset category are provided in Supplementary. We provide some more details in Tab. 1 and Tab. 2 about the training and test datasets, respectively. These details include information like Class, number of frames (Fr.), and geometry precision (Prc.). For RWTT train set, 406 meshes among 568 were selected before processing them to point cloud training set. These mesh vertices are originally available in floating-point values, hence, quantization was performed on the vertices before point-sampling on top of the meshes. For KITTI train set we use sequences 00 to 10 with all their frames, while for the KITTI test set we use sequences 11 to 21 with the first 300 frames from each test sequence.

4. ADDITIONAL RD-CURVES

Additional RD curves are also provided here for comparison with state-of-the-art methods, further illustrating the trends summarized in Table 1 of the main paper. Our method consistently outperforms or matches existing approaches in D1-PSNR across all categories. For D2-PSNR, we achieve clear gains in the dense dynamic category, which benefits from our efficient inter-frame modeling. However, in dense static and sparse dynamic categories, D2-PSNR lags slightly—likely due to our design trade-offs that prioritize D1 accuracy and bitrate efficiency. These curves highlight our method's balanced and competitive rate-distortion behavior.

5. REFERENCES

[1] MPEG Technical Requirements (WG 02), "Updated call for proposals for ai-based point cloud coding," July, 2024, Accessed on Nov 14, 2024.

[2] Andrea Maggiordomo, Federico Ponchio, Paolo Cignoni, and Marco Tarini, "Real-world textured things: A repository of textured models generated with modern photo-reconstruction tools," *Computer Aided Geometric Design*, vol. 83, pp. 101943, 2020.

[3] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.