Beyond Perspective: Neural 360-Degree Video Compression

Supplementary Material

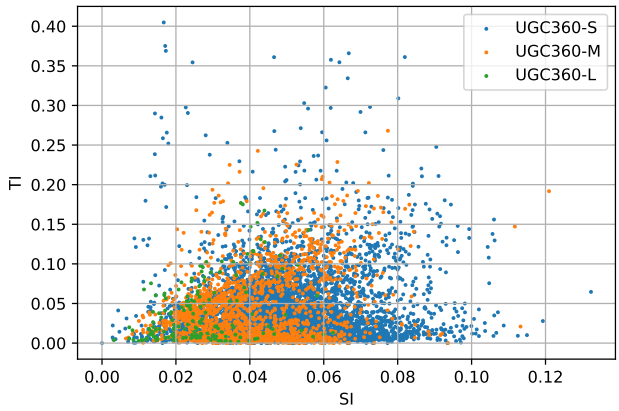

Figure 1. Spatial Information (SI) and Temporal Information (TI) for our UGC360 dataset.

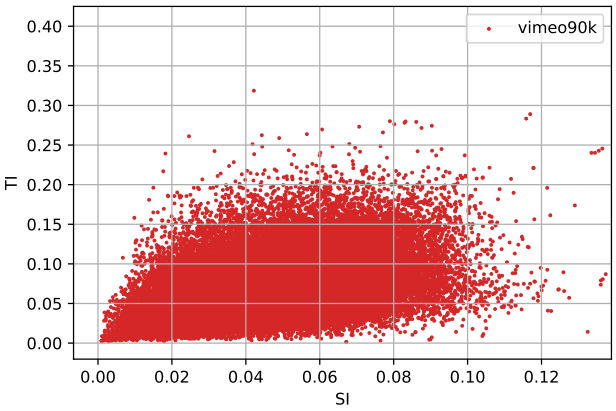

Figure 2. Spatial Information (SI) and Temporal Information (TI) for the vimeo90k [12] dataset.

Subset	Resolutions	Minimum	Maximum	Clips
UGC360-S	0.5K	640 × 320	640 × 360	11
	1K	960 × 480	1280 × 720	398
	1.5K	1430 × 1080	1430 × 1080	5
	2K	1920 × 960	2048 × 1056	4665
	2.5K	2560 × 1440	2560 × 1440	1
UGC360-M	3K	3072 × 1536	3072 × 1536	161
	4K	3840 × 1920	3840 × 2160	1449
	5K	5376 × 2688	5376 × 2688	2
	6K	5760 × 2880	5760 × 2880	6
UGC360-L	8K	7680 × 3840	7680 × 4320	168

Table 1. Distribution of resolutions in the UGC360 dataset.

1. Details on UGC360

Fig. 1 shows the Spatial Information (SI) and Temporal Information (TI) [4] of all clips in the proposed UGC360 dataset. The data points for the different subsets are color-coded as UGC360-S (blue), UGC360-M (orange), and UGC360-L (green). For comparison, Fig. 2 plots the SI/TI for the vimeo90k dataset [12]. Table 1 reports the distribution of resolutions in our UGC360 dataset. Most clips within the UGC360-S subset fall into the 2K resolution category with resolutions ranging from 1920×960 to 2048×1056. In the UGC360-M subset, most clips fall into the 4K resolution category with resolutions ranging from 3840×1920 to 3840×2160. The UGC360-L subset consists exclusively of 8K clips with resolutions ranging from 7680×3840 to 7680×4320.

Each UGC360 subset includes a summary table in csv format with columns: video_id, clip_id, publisher, license, url, published_at. Each row in the table refers to one clip in the dataset. The video_id refers to the id of the video a clip has been extracted from, the clip_id refers to the corresponding clip number. As clips are de-duplicated via their gist descriptors and clips with low contrast are removed, there might be gaps despite incremental clip numbering. Publisher refers to the platform a clip has been obtained from. License describes the CC license of the corresponding video. The URL refers to the original video URL that could, e.g., be used to extract longer clips for the videos in the dataset.

2. Details on Mipmapping

Mipmapping is a well-known technique from computer graphics that is utilized if sampling densities between source and target meshes vary significantly among different mesh positions [11]. Due to the vastly different scales at which the pixel values are sampled in the ERP format - the spherical sampling density increases significantly towards the poles - reprojection may cause severe aliasing if polar regions in the source format are projected to less densely sampled regions in the target projection, e.g., the equator. Mipmapping prepares the original image at multiple scales and dynamically selects the most suitable scale for interpolation of each pixel position. For each mipmap level, the original image is downscaled by a factor of 2 including Gaussian pre-filtering for antialiasing. Fig. 3 shows an image with 8 mipmap levels, which means that the lowest scale is downsampled by a factor of $2^8 = 256$ with respect to the original image. This process allows to efficiently pre-compute the original image at various antialiased scales for fast antialiased interpolation during resampling. The

Table 2. BD-Rate (%) in RGB color space with respect to the baseline DCVC-HEM finetuned on vimeo90k. Different model and training set combinations. WS-PSNR used as quality metric for the 360-degree JVET360 dataset, PSNR for the remaining perspective datasets. Column *Average* shows the average BD-Rate over all perspective dataset sequences. Highest rate savings for each dataset are marked bold.

Model	Training set	FGR	JVET360	HEVC-B	HEVC-C	HEVC-D	HEVC-E	UVG	Average
DCVC-HEM [6]	vimeo90k		0.00	0.00	0.00	0.00	0.00	0.00	0.00
DCVC-HEM [6]	UGC360		-2.87	9.89	0.93	6.69	-5.39	8.43	5.61
		✓	-6.68	3.69	-2.84	4.76	-11.17	2.55	0.71
DCVC-HEM [6]	UGC360+ vimeo90k		-3.69	2.33	5.23	4.10	-7.43	0.19	1.13
		✓	-6.21	-1.21	1.39	1.52	-10.88	-1.69	-1.82
DCVC-HEM with PFE	UGC360+ vimeo90k		-5.31	4.36	1.28	1.93	-6.30	0.80	0.98
		✓	-7.74	-1.95	-1.60	-2.62	-13.10	-3.94	-4.08
HM-18.0 [10]			30.34	30.34	3.82	23.27	12.13	24.05	20.45
VTM-22.2 [2]			-3.61	-5.85	-24.00	-7.45	-23.16	-6.40	-11.66

mipmap level (Level of Detail, LoD) to use for interpolating a pixel value is then obtained as follows [8]. The Jacobian of the target coordinates in the source domain $\boldsymbol{p}_{\text{tar}\to\text{src}} = (u_{\text{tar}\to\text{src}}, v_{\text{tar}\to\text{src}})^{\text{T}}$ with respect to the target coordinates $\boldsymbol{p}_{\text{tar}} = (u_{\text{tar}}, v_{\text{tar}})^{\text{T}}$ is calculated for each target pixel position $\boldsymbol{p}_{\text{tar}} \in \mathcal{P}_{\text{tar}}$

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial u_{\text{tar}\to\text{src}}}{\partial u_{\text{tar}}} & \frac{\partial u_{\text{tar}\to\text{src}}}{\partial v_{\text{tar}}} \\ \frac{\partial v_{\text{tar}\to\text{src}}}{\partial u_{\text{tar}}} & \frac{\partial v_{\text{tar}\to\text{src}}}{\partial v_{\text{tar}}} \end{bmatrix}. \tag{1}$$

This matrix describes how a small step in the target domain (reprojected image) translates to the source domain (original image). An infinitesimal step along the u_{tar}-axis in the target domain consequently yields a displacement by

$$S_u = \sqrt{\left(\frac{\partial u_{\text{tar}\to\text{src}}}{\partial u_{\text{tar}}}\right)^2 + \left(\frac{\partial v_{\text{tar}\to\text{src}}}{\partial u_{\text{tar}}}\right)^2} \tag{2}$$

in the source domain. Similarly, an infinitesimal step along the v_{tar}-axis in the target domain yields a displacement by

$$S_v = \sqrt{\left(\frac{\partial u_{\text{tar}\to\text{src}}}{\partial v_{\text{tar}}}\right)^2 + \left(\frac{\partial v_{\text{tar}\to\text{src}}}{\partial v_{\text{tar}}}\right)^2} \tag{3}$$

in the source domain. These scales S_u and S_v represent the size of a target pixel (reprojected image) in the source domain (original image). Typically, the maximum scale is considered [8]

$$S = \max(S_u, S_v). \tag{4}$$

The LoD is then selected such that one mipmapped pixel closely matches the scaled target pixel size. Because the mipmap resolution is halved in each step (i.e., the relative mipmapped pixel size is doubled in each step), the LoD is obtained as

$$\text{LoD} = \lfloor \log_2(S) \rfloor. \tag{5}$$

In most computer graphics applications, a weighted average of the obtained pixel values from both surrounding mipmap levels is taken. In the regarded 360-degree data augmentation scenario, only the higher resolution mipmap level is selected in favor of reduced computational complexity. This is realized by flooring in (5). The LoD is calculated for each target pixel position individually, such that interpolation of each target pixel value is performed on the appropriate mipmap level.

Figure 3. Mipmapped image with 8 mipmap levels. For each mipmap level, the original image is downscaled by a factor of 2 including Gaussian pre-filtering for antialiasing.

3. Further Results

3.1. RGB Color Space

Table 2 shows the rate savings achieved by our 360-degree optimization framework for DCVC-HEM [6] and the extended DCVC-HEM-360 in RGB color space. It validates the results from the main paper, where our proposed framework was evaluated in YUV color space.

Table 3. Ablation study investigating extension of different network modules with positional feature encoding. BD-Rate (%) in YUV color space with respect to default configuration M_A. All instances trained on UGC360+vimeo90k with flow-guided reprojection.

	M_A	M_B	M_C	M_D	M_E
Contextual encoder		✓	✓	✓	✓
Contextual decoder			✓	✓	✓
Entropy model	✓			✓	✓
Context generator					✓
JVET360	0.00	0.24	0.32	0.42	-0.33
HEVC+UVG	0.00	0.27	4.13	1.13	3.29
Million parameters	17.528	17.524	17.527	17.532	17.532
GMACs	872.01	872.08	872.39	872.39	872.43

3.2. Positional Feature Encoding

Table 3 shows the results of an ablation study investigating the influence of the proposed positional feature encoding. We train different variants M_A - M_E of DCVC-HEM-360 incorporating the positional feature encoding at different positions in the network. Model M_A denotes the default configuration, where the positional feature encoding is only introduced to the entropy model. All models are trained on the combined dataset UGC360+vimeo90k with flow-guided reprojection.

While the performance between the different models varies only slightly for 360-degree video - model M_E even achieves slight rate savings by 0.33% over the default model - significant performance differences occur for perspective video. Introducing the positional feature encoding into all major network components leads to an increase in rate by more than 3% for perspective video. From a theoretical perspective, models M_A - M_D are a special case of model M_E, where the respective kernel weights for the additional positional feature encoding channel are set to 0. However, as training of neural networks is not a convex optimization problem, it can not be guaranteed that training converges to a global optimum. Thus, as the results show, it can be the better option to omit additional parameters if they do not prove to be beneficial. An additional benefit is the reduced complexity overhead.

3.3. Performance with Other Models

Tables 4 and 5 show the achieved rate savings on 360-degree and perspective video using the DCVC [5] and DCVC-TCM [9] models in YUV and RGB color space, respectively. DCVC-360 and DCVC-TCM-360 refer to the extended NVCs incorporating positional feature encoding into the respective entropy models as described in Section 3.3 in the main paper. All models are finetuned on the combined vimeo90k+UGC360 dataset using flow-guided reprojection for data augmentation. To ensure a fair evaluation, the baseline models of DCVC and DCVC-TCM are finetuned on vimeo90k for the same number of iterations as the models trained on vimeo90k+UGC360. For DCVC, training on the combined dataset with flow-guided reprojection yields average rate savings of 6.05% (4.95%) for 360-degree video and 3.65% (2.69%) for perspective video in YUV (RGB) color space. For DCVC-360 with positional feature encoding, rate savings increase to 8.07% (7.51%) for 360-degree video and 4.93% (4.07%) for perspective video.

Table 4. Evaluation of the proposed 360-degree NVC framework for the DCVC and DCVC-TCM models. BD-Rate (%) in YUV color space with respect to DCVC trained on vimeo90k for DCVC and DCVC-360, and with respect to DCVC-TCM trained on vimeo90k for DCVC-TCM and DCVC-TCM-360.

Model	JVET360	Average
DCVC [5]	-6.05	-3.65
DCVC-360	**-8.07**	**-4.93**
DCVC-TCM [9]	-4.25	2.87
DCVC-TCM-360	**-4.81**	**0.24**

Table 5. Evaluation of the proposed 360-degree NVC framework for the DCVC and DCVC-TCM models. BD-Rate (%) in RGB color space with respect to DCVC trained on vimeo90k for DCVC and DCVC-360, and with respect to DCVC-TCM trained on vimeo90k for DCVC-TCM and DCVC-TCM-360.

Model	JVET360	Average
DCVC [5]	-4.95	-2.69
DCVC-360	**-7.51**	**-4.07**
DCVC-TCM [9]	-4.55	1.98
DCVC-TCM-360	**-4.91**	**-0.12**

For DCVC-TCM, training on the combined dataset with flow-guided reprojection yields average rate savings of 4.25% (4.45%) for 360-degree video, but increases in rate by 2.87% (1.98%) for perspective video. With rate savings of 4.81% (4.91%) for 360-degree video, DCVC-TCM-360 with positional feature encoding improves only slightly over DCVC-TCM trained with the combined dataset and flow-guided reprojection. However, the losses experienced for perspective video are significantly reduced. In YUV color space, the increase in rate is significantly reduced from 2.87% (DCVC-TCM) to 0.24% (DCVC-TCM-360). In RGB color space, slight rate savings of 0.12% are achieved.

These results validate that our approach generates improvements in 360-degree video compression performance for other NVC architectures as well, while retaining compression performance for traditional perspective video.

3.4. Rate-Distortion Curves

Fig. 4 - 9 show the Rate-Distortion curves (RD curves) for all sequences in the JVET360 [3], HEVC-B, -C, -D, -E [1], and UVG [7] datasets. RD curves for the baseline DCVC-HEM finetuned on vimeo90k are shown in blue, RD curves for DCVC-HEM finetuned on the combined UGC360+vimeo90k without FGR are shown in orange, RD curves for DCVC-HEM finetuned on UGC360+vimeo90k with FGR are shown in green, and RD curves for the extended DCVC-HEM-360 with positional feature encoding finetuned on UGC360+vimeo90k are shown in red. For context, the RD curves for the traditional HEVC [10] and VVC [2] are shown with dashed lines in purple and brown.

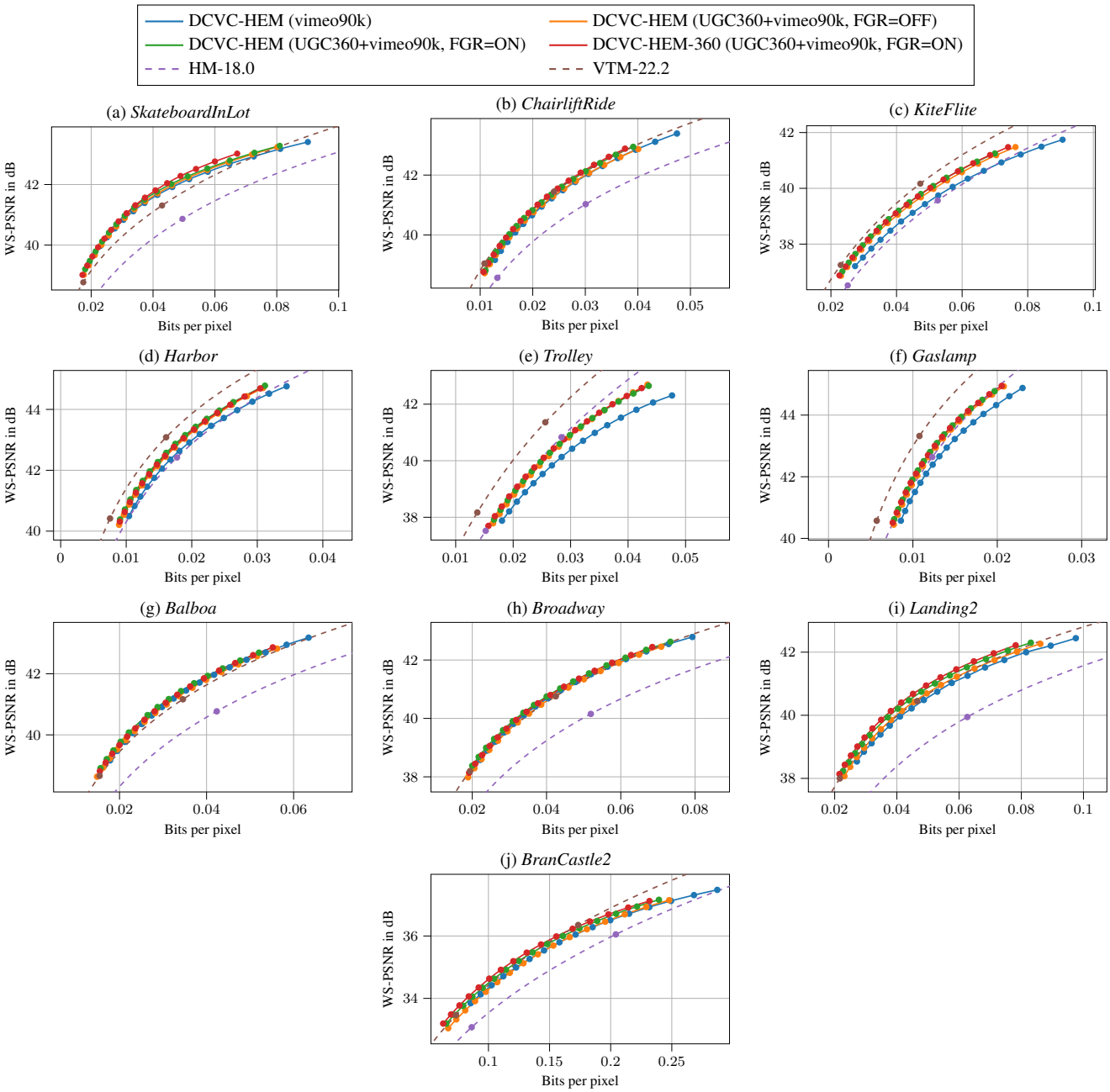

Figure 4. RD curves in YUV color space for each sequence in the JVET360 dataset.

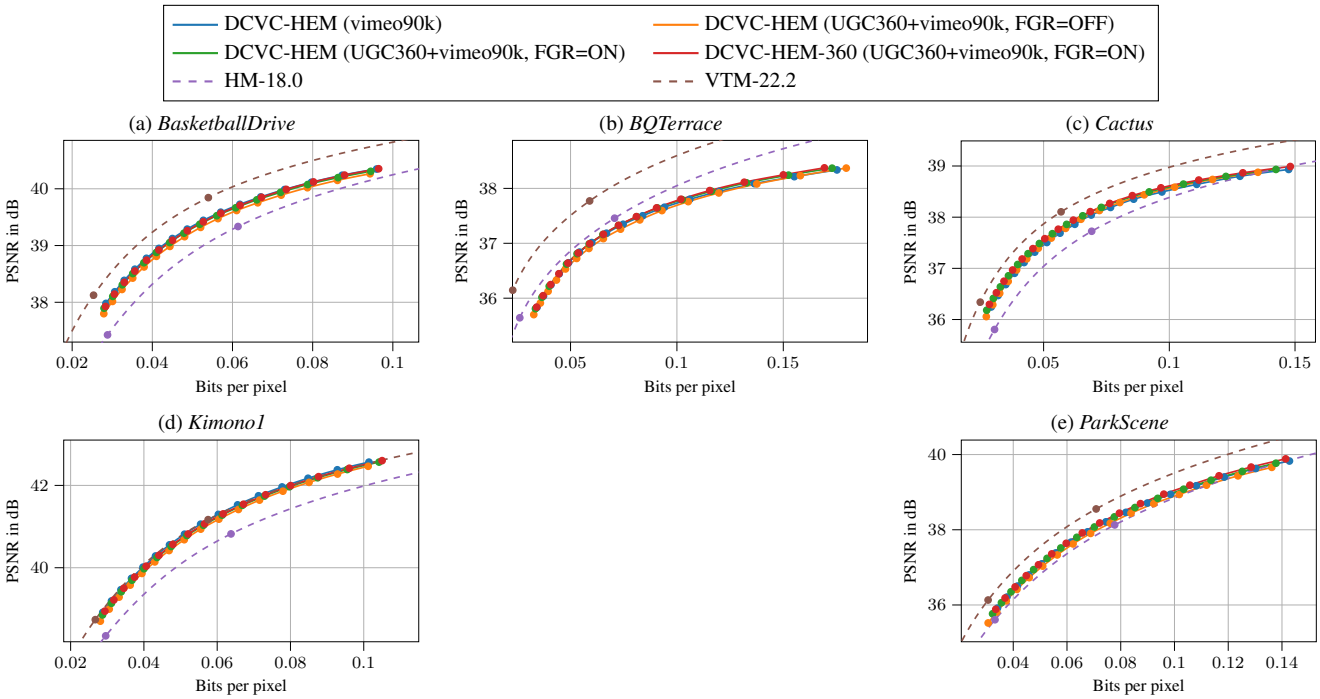

Figure 5. RD curves in YUV color space for each sequence in the HEVC-B dataset.

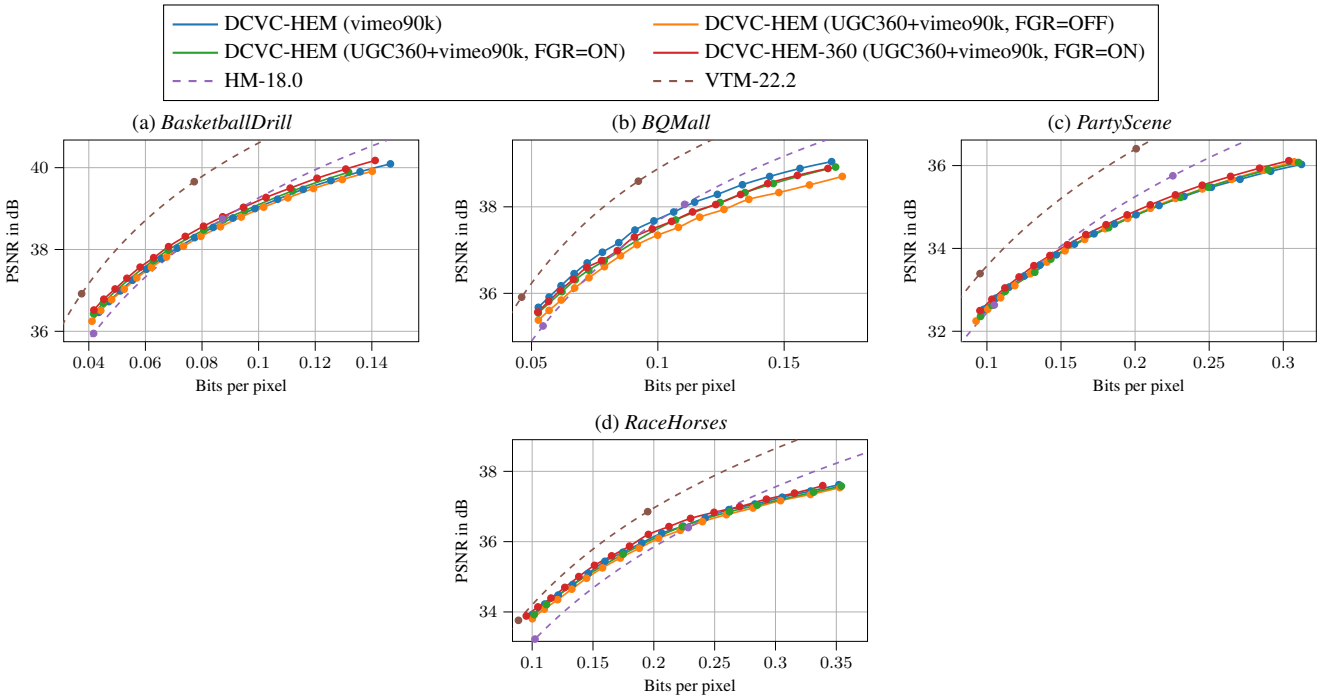

Figure 6. RD curves in YUV color space for each sequence in the HEVC-C dataset.

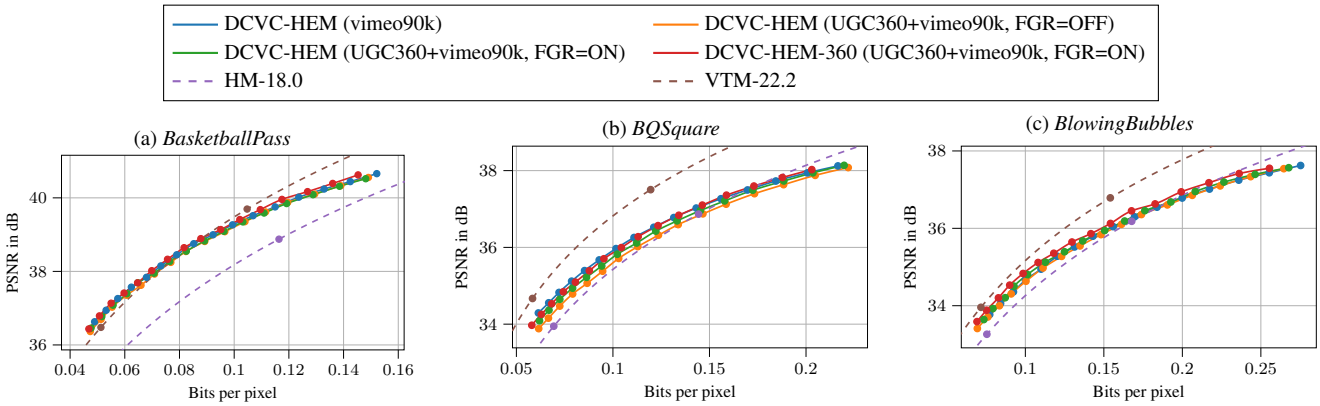

Figure 7. RD curves in YUV color space for each sequence in the HEVC-D dataset.

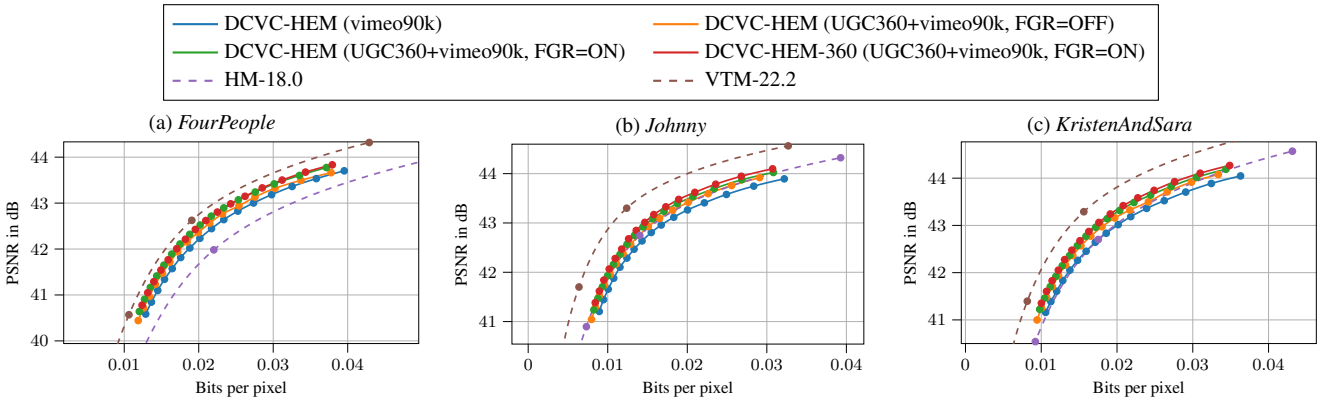

Figure 8. RD curves in YUV color space for each sequence in the HEVC-E dataset.

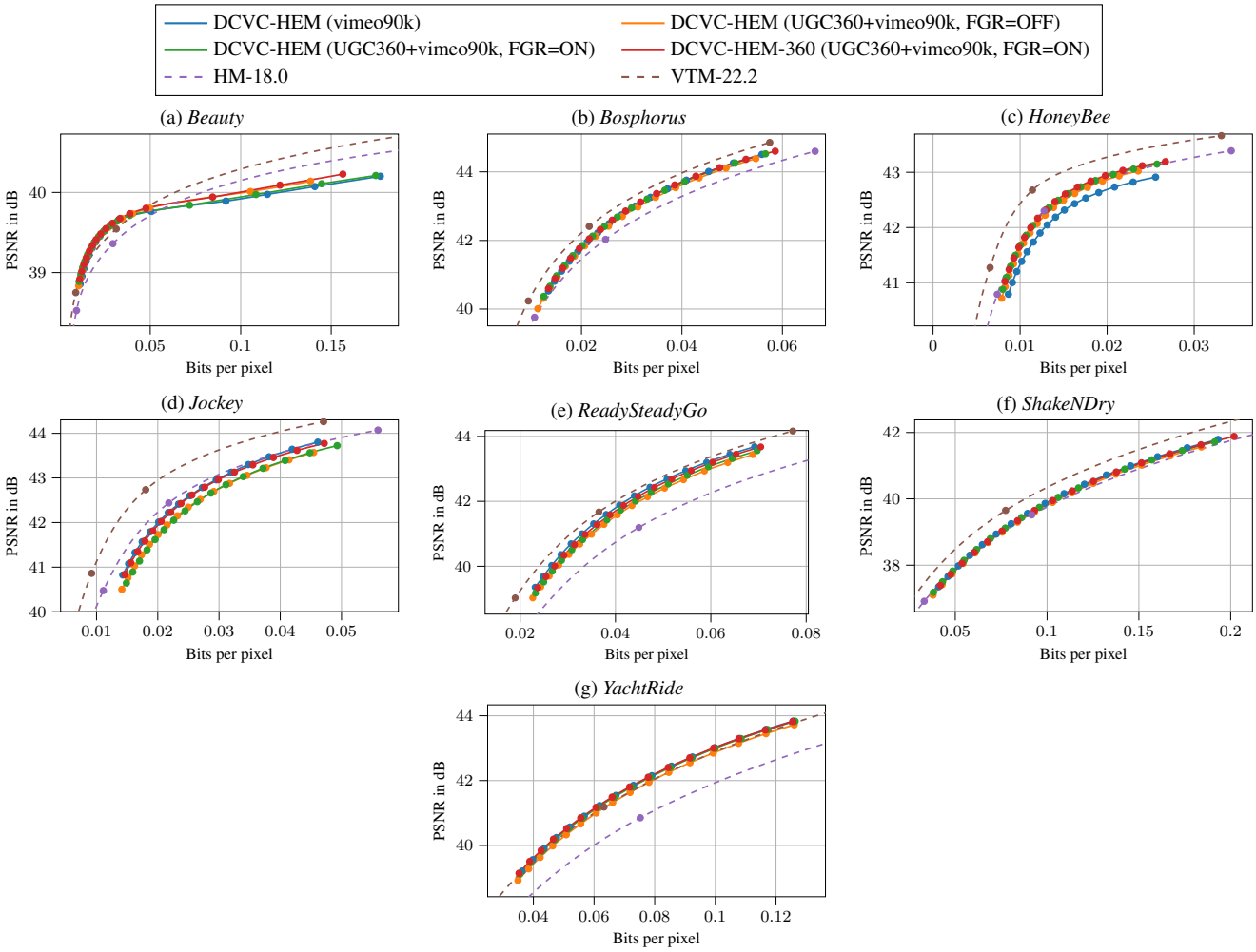

Figure 9. RD curves in YUV color space for each sequence in the UVG dataset.

References

[1] Frank Bossen, Jill Boyce, Karsten Sühring, Xiang Li, and Vadim Seregin. VTM Common Test Conditions and Software Reference Configurations for SDR Video, JVET-T2010. In *Proc. 20th Meet. Jt. Video Experts Team*, pages 1–2, 2020. 4

[2] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the Versatile Video Coding (VVC) Standard and its Applications. *IEEE Trans. Circuits Syst. Video Technol.*, 31(10): 3736–3764, 2021. 2, 4

[3] Yuwen He, Jill Boyce, Kiho Choi, and Jian-Liang Lin. JVET Common Test Conditions and Evaluation Procedures for 360° Video, JVET-U2012. In *Proc. 21st Meet. Jt. Video Explor. Team*, pages 1–8, 2021. 4

[4] ITU-T. ITU-T Rec. P.910: Subjective Video Quality Assessment Methods for Multimedia Applications, 2023. 1

[5] Jiahao Li, Bin Li, and Yan Lu. Deep Contextual Video Compression. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 18114–18125, 2021. 3

[6] Jiahao Li, Bin Li, and Yan Lu. Hybrid Spatial-Temporal Entropy Modelling for Neural Video Compression. In *Proc. 30th ACM Int. Conf. Multimed.*, pages 1503–1511, 2022. 2

[7] Alexandre Mercat, Marko Viitanen, and Jarno Vanne. UVG Dataset: 50/120fps 4K Sequences for Video Codec Analysis and Development. In *Proc. 11th ACM Multimed. Syst. Conf.*, pages 297–302, 2020. 4

[8] Mark Segal and Kurt Akeley. The OpenGL Graphics System: A Specification - Version 4.6 (Core Profile). pages 1–829, 2022. 2

[9] Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu. Temporal Context Mining for Learned Video Compression. *IEEE Trans. Multimed.*, 25:7311–7322, 2023. 3

[10] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Trans. Circuits Syst. Video Technol.*, 22(12):1649–1668, 2012. 2, 4

[11] Lance Williams. Pyramidal Parametrics. *SIGGRAPH Comput. Graph.*, 17(3):1–11, 1983. 1

[12] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman. Video Enhancement with Task-Oriented Flow. *Int J Comput Vis*, 127(8):1106–1125, 2019. 1