Deep Learning in Latent Space for Video Prediction and Compression Supplementary Materials

Bowen Liu Yu Chen Shiyu Liu Hun-Seok Kim
University of Michigan, Ann Arbor
{bowenliu, unchenyu, shiyuliu, hunseok}@umich.edu

1. Model Architecture

The detailed neural network structure of our predictor and decoder model is shown in Figure 1 and Figure 2.

1.1. Prediction network

Figure 1 shows our predictor model architecture in the proposed video compression codec. The kernel size of all convolutional layers in *ResBlocks* and *ConvLSTM* layers is 3×3. The first *ResBlock* has 64 filters in its convolutional layers, other *ResBlocks* have 128 filters in their convolutional layers. We adopt the *ConvLSTM* layer used in [6].

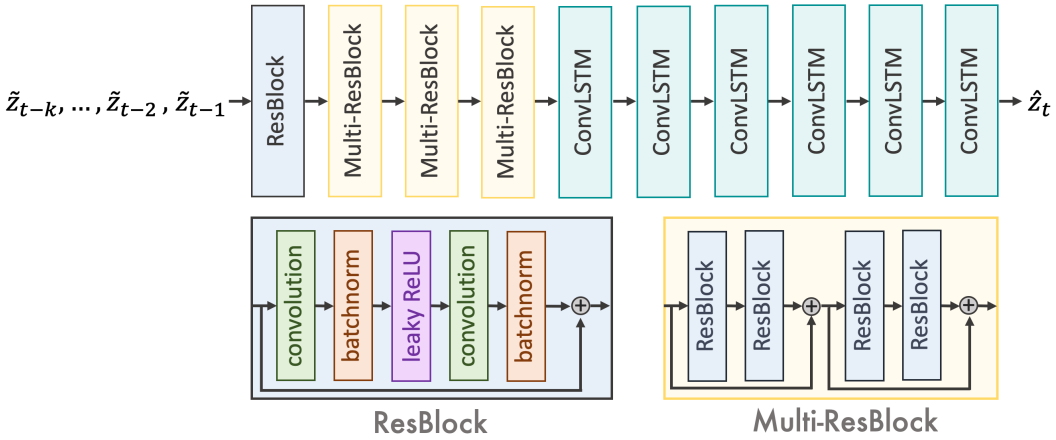


Figure 1: Prediction network structure.

1.2. Decoder network

Figure 2 shows the decoder network to generate video frames from latent vectors. The kernel size of transpose convolutional layers in the *TransConv+ReLU* block and those in all *ResBlocks* except for the last two is 3×3. The last two *ResBlocks* have 5×5 kernels. The transpose convolutional layer of the last two *ResBlock* has 64 and 3 filters respectively. The number of filters is 128 in the transpose convolutional layers elsewhere.

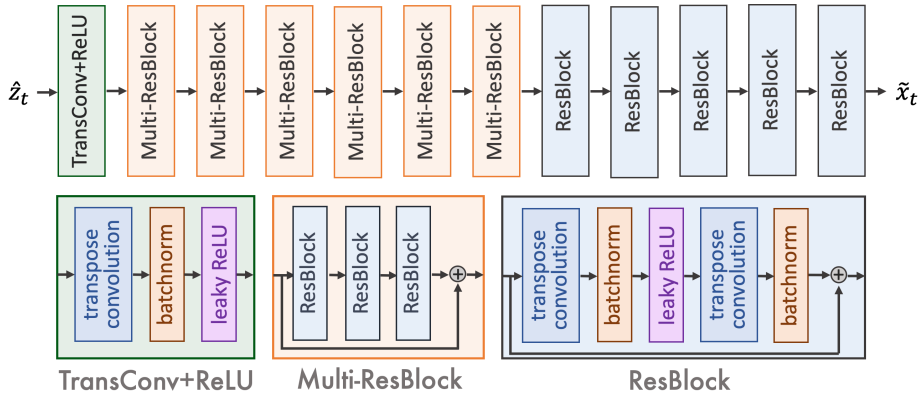


Figure 2: Decoder network structure.

2. AVC/H.264 and HEVC/H.265 configurations

Following the prior work [5], we use *ffmpeg* to produce baseline results for AVC/H.264 and HEVC/H.265 codecs. Videos for PSNR/SSIM measurements are encoded by *ffmpeg* very-fast mode with a GOP of 10/12 for the VTL/UVG dataset. For qualitative comparisons, we encode the videos by *ffmpeg* fast mode to produce baseline videos with similar bit rate as ours. Applied *ffmpeg* settings are listed as follows:

AVC/H.264:

```
ffmpeg -y -pix_fmt yuv420p -s [HxW] -r [Frame Rate] -i [Target File].yuv
-vframes [Nframes] -c:v libx264 -preset [Mode] -tune zerolatency -crf [CRF]
-g [GOP] -bf 2 -b_strategy 0 -sc_threshold 0 -loglevel debug
[Dest File].mkv
```

hevc/H.265:

```
ffmpeg -pix_fmt yuv420p -s [HxW] -r [Frame Rate] -i [Target File].yuv
-vframes [Nframes] -c:v libx265 -preset [Mode] -tune zerolatency -crf [CRF]
-g [GOP] [Dest File].mkv
```

3. Subjective frame quality study

In this section, we show qualitative comparison between results of the proposed method and HEVC/H.265 with frames in the UVG and VTL datasets. Our method outperforms HEVC/H.265 on both datasets with better handled details, less artifacts, and superior color quality in real world video sequence. All figures are presented with their original resolution. Note that the quality difference for samples from the UVG dataset can be better viewed by zooming-in the PDF.

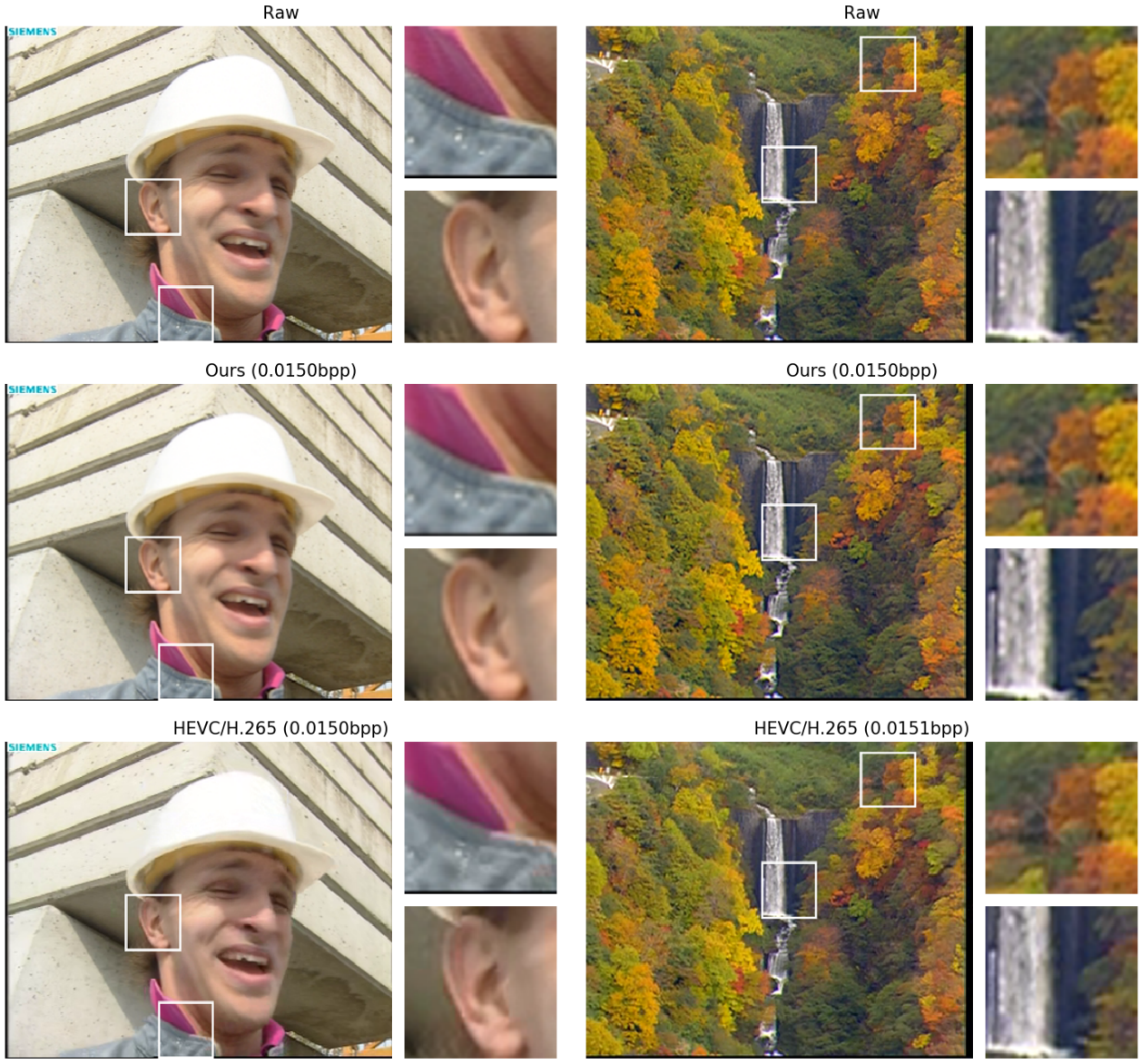


Figure 3: Subjective comparison on example frames from the VTL dataset.

Figure 4: Subjective comparisons on a frame from the UVG dataset. Compared to the HEVC/H.265 coded frame, out method has better quality on details (see the content nearby the white barriers). Out method also retains the boundaries between objects better (see the rider's helmet and costume).

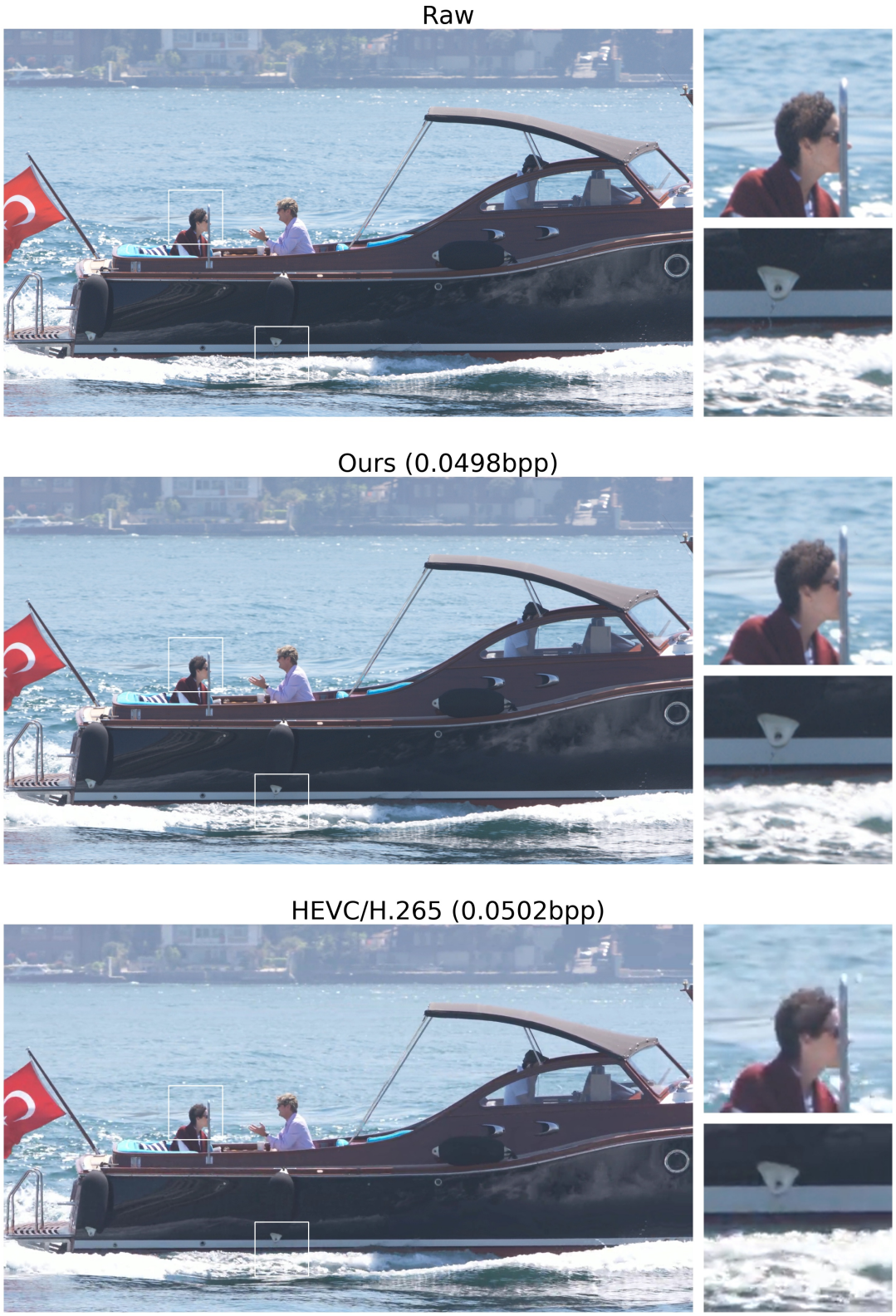
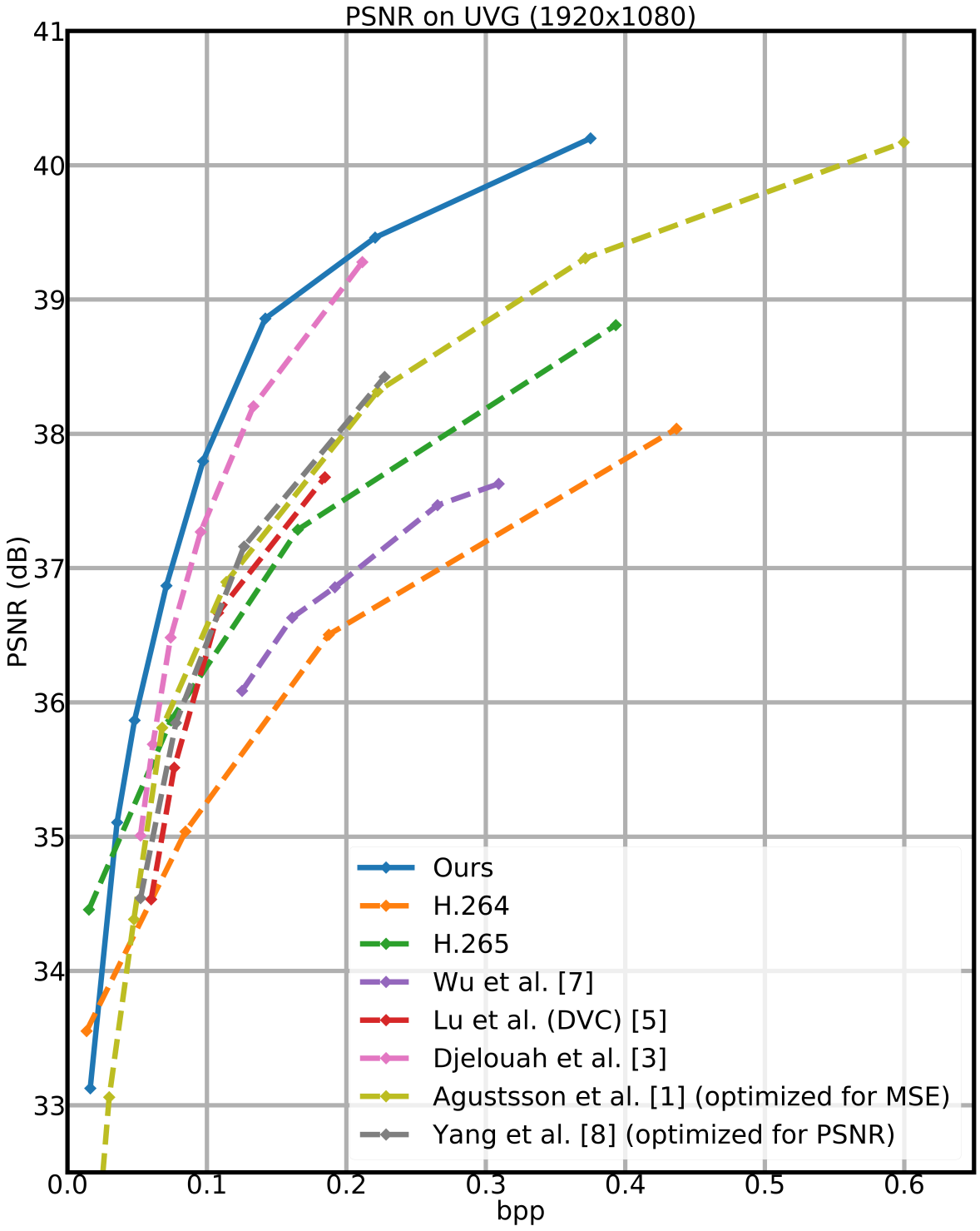


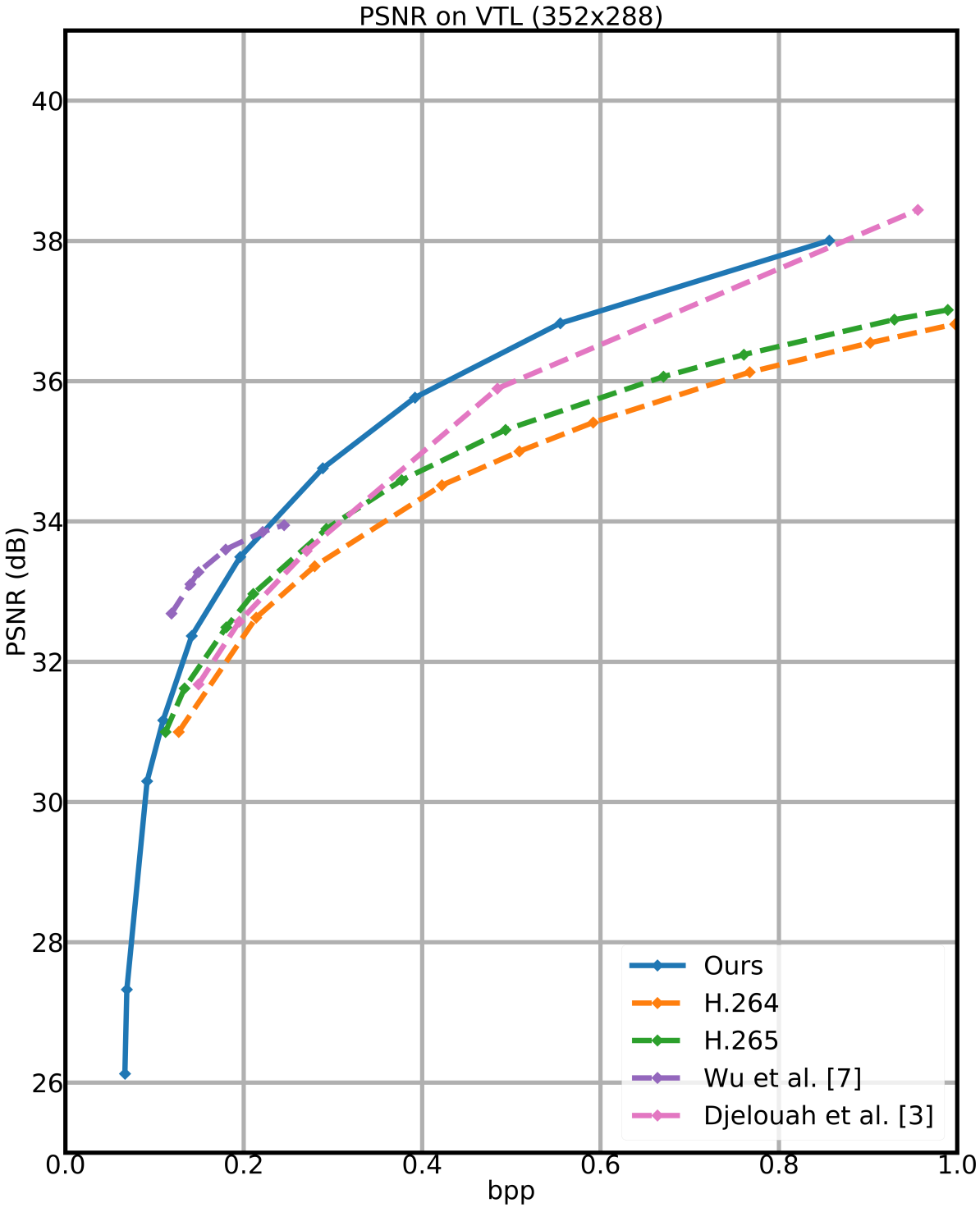
Figure 5: Subjective comparisons on an example frame from the UVG dataset. Regarding color fidelity, our approach maintains the original tones while HEVC/H.265 shows a slightly faded tone with less refined contents (see the waves and the passenger's face).

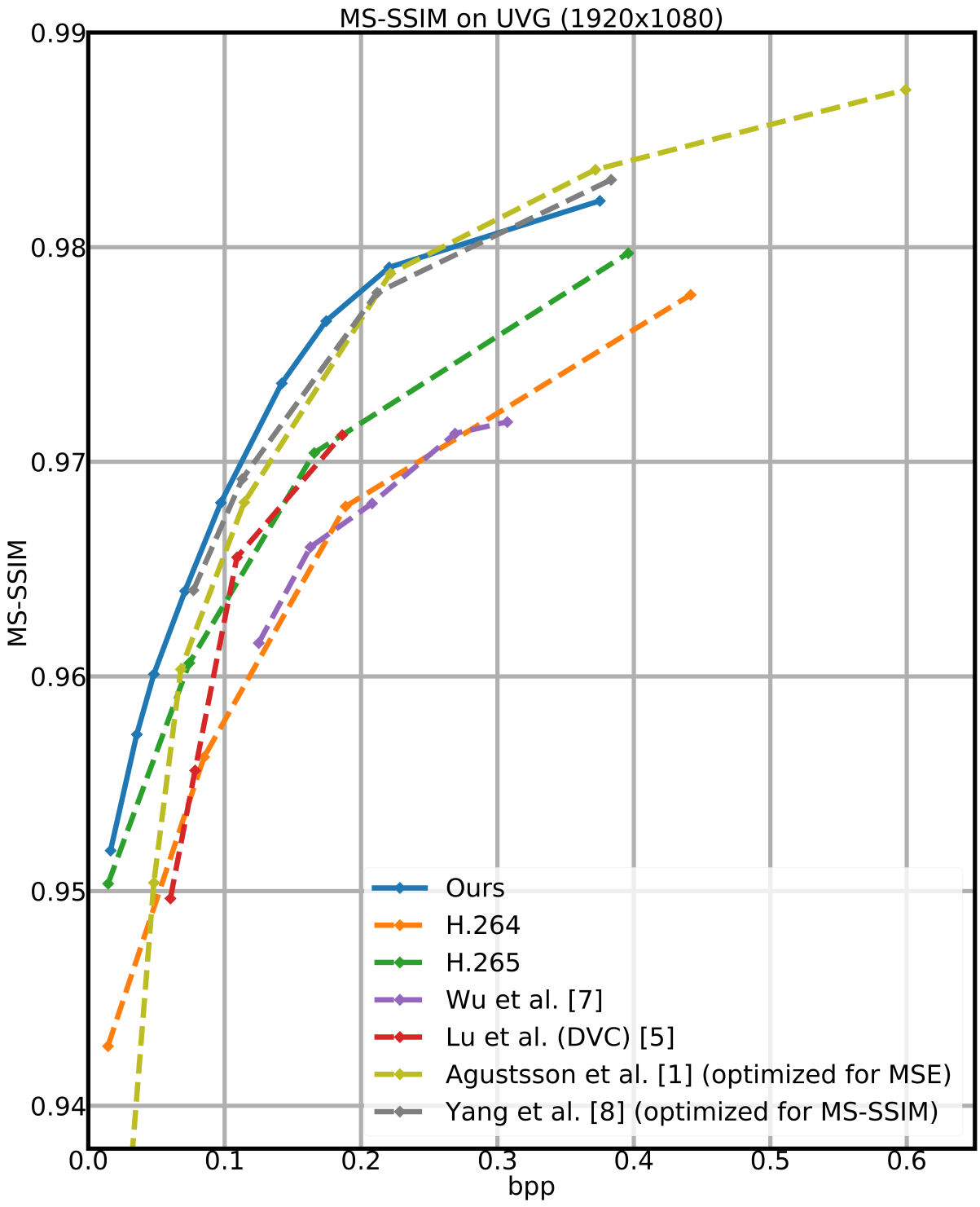
Figure 6: Subjective comparisons on an example frame from the UVG dataset. Our method provides sharper details compared to HEVC/H.265 (see regions around the horse's eye and the edges of the saddle pad).

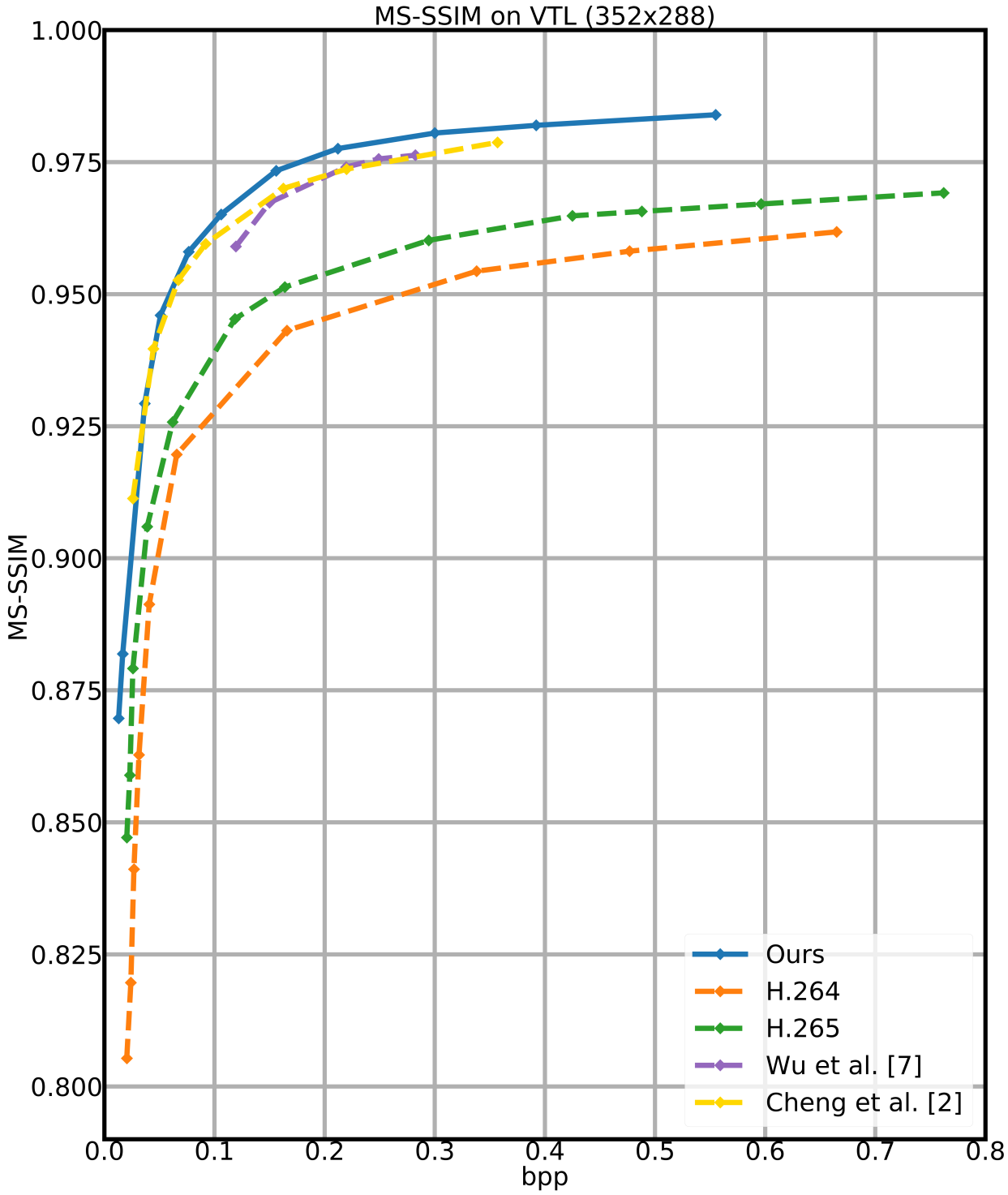
4. Aggregated rate-distortion curves

The size of the rate-distortion curves provided in the manuscript was constrained by the available space. Here we provide identical but larger rate-distortion curves for enhanced readability to compare our work with [7, 5, 2, 3, 4, 1, 8].




PSNR on VTL (352x288)
PSNR (dB)
bpp
40
38
36
34
32
30
28
26
0.0
0.2
0.4
0.6
0.8
1.0
Ours
H.264
H.265
Wu et al. [7]
Djelouah et al. [3]


MS-SSIM on UVG (1920x1080)
MS-SSIM
bpp
0.99
0.98
0.97
0.96
0.95
0.94
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Ours
H.264
H.265
Wu et al. [7]
Lu et al. (DVC) [5]
Agustsson et al. [1] (optimized for MSE)
Yang et al. [8] (optimized for MS-SSIM)


MS-SSIM on VTL (352x288)
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0.800
MS-SSIM
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
bpp
Ours
H.264
H.265
Wu et al. [7]
Cheng et al. [2]

References

[1] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 7

[2] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learning image and video compression through spatial-temporal energy compaction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[3] Abdelaziz Djelouah, Joaquim Campos, Simone Schaub-Meyer, and Christopher Schroers. Neural inter-frame compression for video coding. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 7

[4] Amirhossein Habibian, Ties van Rozendaal, Jakub M. Tomczak, and Taco S. Cohen. Video compression with rate-distortion autoencoders. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 7

[5] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 7

[6] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc., 2015. 1

[7] Chao-Yuan Wu, Nayan Singhal, and Philipp Krähenbühl. Video compression through image interpolation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, volume 11212 of *Lecture Notes in Computer Science*, pages 425–440. Springer, 2018. 7

[8] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with hierarchical quality and recurrent enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 7