Ultra-Fast Neural Video Compression

Supplementary Material

This document provides supplementary material for our paper, detailing the experimental settings and additional results for our proposed ultra-fast neural video codec (NVC), DCVC-UF (Ultra-Fast).

1. Test Settings

This section details the experimental configurations used for comparing our NVC with traditional codecs across both YUV420 and RGB (shown in Section 3) colorspaces.

YUV420 Colorspace. The YUV420 colorspace is the standard for most practical video applications and traditional codecs, which are highly optimized for this format. Therefore, benchmarking in YUV420 is crucial for assessing the practical performance of NVCs against established standards. For traditional codecs, we use the reference software for H.265 (HM [1]) and H.266 (VTM [2]). In the low-delay setting, we use the *encoder_lowdelay_main10.cfg* and *encoder_lowdelay_vtm.cfg* configurations for HM and VTM, respectively. For the delay-relaxed setting, we use *encoder_randomaccess_main10.cfg* and *encoder_randomaccess_vtm.cfg*. The parameters for each video are as follows:

- -c {*config file name*}
  --InputFile={*input video name*}
  --InputBitDepth=8
  --OutputBitDepth=8
  --OutputBitDepthC=8
  --FrameRate={*frame rate*}
  --DecodingRefreshType=2
  --FramesToBeEncoded={*frame number*}
  --SourceWidth={*width*}
  --SourceHeight={*height*}
  --IntraPeriod={*intra period*}
  --QP={*qp*}
  --Level=6.2
  --BitstreamFile={*bitstream file name*}

For both low-delay and delay-relaxed settings, it should be noted that we set the intra-period to –1 for HM and VTM to report their best compression ratio. Here we show a comparison for VTM under the delay-relaxed setting: if using VTM with intra-period=32 as the anchor, VTM with intra-period=–1 can achieve an average of 19.8% bitrate saving on the six test datasets (HEVC Class B∼E, UVG, and MCL-JCV).

RGB Colorspace. Since our test datasets are originally in the YUV420 format, we convert them to RGB for this evaluation. Following the methodology of JPEG AI [3, 4] and [6], we use the BT.709 standard for the YUV-to-RGB conversion. This is because using BT.709 obtains higher compression ratio under the similar visual quality when compared with the BT.601. As demonstrated in [6], traditional codecs achieve superior compression when encoding RGB content by using an internal 10-bit YUV444 pipeline, even though the final distortion is measured in RGB. We adopt this same best-practice configuration for our tests. For HM and VTM, we use the *encoder_lowdelay_main_rext.cfg* and *encoder_lowdelay_vtm.cfg* configurations for the low-delay scenario. The parameters for each video are as follows:

- -c {*config file name*}
  --InputFile={*input file name*}
  --InputBitDepth=10
  --OutputBitDepth=10
  --OutputBitDepthC=10
  --InputChromaFormat=444
  --FrameRate={*frame rate*}
  --DecodingRefreshType=2
  --FramesToBeEncoded={*frame number*}
  --SourceWidth={*width*}
  --SourceHeight={*height*}
  --IntraPeriod={*intra period*}
  --QP={*qp*}
  --Level=6.2
  --BitstreamFile={*bitstream file name*}

For both YUV420 and RGB evaluations, all traditional codecs were configured with their best settings and reference structures to ensure rigorous comparison.

2. Depth-wise Convolution Block Details

Our DCVC-UF architecture is deliberately built from a single, lightweight primitive: depth-wise convolution (DC) blocks (as illustrated in Fig. 1) [5], which serve as the fundamental building units across all modules. By avoiding complex operations or heterogeneous specialized blocks, the overall network remains conceptually simple and highly efficient to implement. For the delay-relaxed setting with chunk size $N = 8$, our high-throughput model DCVC-UF (HT-S) employs 6, 7, and 11 DC blocks in the chunk encoder, chunk decoder, and cross-chunk context generation modules, respectively. Each of the 8 frame-specific decoders (corresponding to the 8 frames in the chunk) contains 3 DC blocks, enabling all frames to be reconstructed in parallel. The larger configuration DCVC-UF (HT-L) increases the numbers of DC blocks to 7, 11, and 12 for the chunk encoder, chunk decoder, and cross-chunk context generation modules, respectively, and allocates 5 DC blocks to each of the 8 frame-specific decoders, providing stronger modeling

Table 1. BD-Rate (%) comparison in RGB colorspace. All frames are tested.

Method	UVG	MCL-JCV	HEVC B	HEVC C	HEVC D	HEVC E	Average	Coding Speed	
								Enc.	Dec.
Low-Delay (LD) Codecs									
VTM-17.0 (LD)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.01 FPS	23.6 FPS
HM-16.25 (LD)	43.2	49.5	49.9	45.2	39.9	47.7	45.9	0.05 fps	39.6 fps
DCVC-DC	9.2	0.0	14.9	5.3	−7.8	87.7	18.2	2.3 FPS	2.9 FPS
DCVC-FM	−10.4	−1.1	−11.2	−26.5	−33.7	−12.1	−15.8	3.7 FPS	4.4 FPS
DCVC-RT	−17.2	−6.8	−11.3	−15.8	−21.3	−11.4	−14.0	118.8 FPS	105.3 FPS
DCVC-UF (LD)	−7.2	8.6	2.2	−1.5	−10.3	−4.0	−2.0	**313.6** FPS	**353.8** FPS
Delay-Relaxed Codecs									
DCVC-UF (HT-S)	−22.7	−3.1	−13.0	−24.8	−37.3	−52.7	−25.6	**655.9** FPS	**453.3** FPS
DCVC-UF (HT-L)	−34.5	−16.2	−29.4	−37.4	−47.5	−57.7	−37.1	**371.1** FPS	**273.6** FPS

Intra-period=–1 for all codecs and settings. The coding speeds of NVCs are tested on 1920×1080 videos with 4090 GPU.

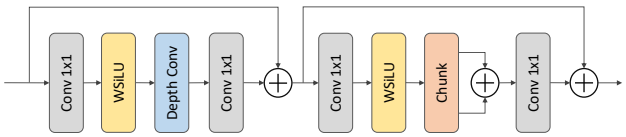

Figure 1. The structure of DC Block (depth-wise convolution block) [5].

capacity at modest additional complexity. For the low-delay setting with chunk size $N = 1$, DCVC-UF (LD) adopts a more compact design with 3, 3, and 9 DC blocks in the chunk encoder, chunk decoder, and cross-chunk context generation modules, respectively, and a single frame decoder comprising 3 DC blocks, yielding an efficient instantiation tailored to low-delay applications.

3. Results on RGB colorspace

Our DCVC-UF framework also employs a unified YUV444 colorspace that serves as a universal interface for both input and output. For both YUV420 and RGB videos, they will be converted into YUV444 for inference and converted back for loss calculation. The unified approach enhances practical deployment by allowing a single trained model to efficiently process multiple video formats without requiring specialized model variants for different formats, thereby reducing storage requirements and deployment complexity. The model is also trained exclusively on YUV444 colorspace, which greatly simplifies the training process. Table 1 presents the performance comparisons in the RGB colorspace. From this table, we can find that our DCVC-UF models achieve the advanced rate-distortion-complexity trade-off. These results validate the effectiveness of DCVC-UF for practical video compression applications.

4. Rate-Distortion Curves

This section presents comprehensive rate-distortion (RD) curves across multiple datasets in the YUV420 colorspace, with all frames tested under the intra-period=–1 configuration. As illustrated in Fig. 2, our analysis reveals different performance characteristics across different quality ranges and datasets. In the lower quality range, our proposed DCVC-UF (HT-L) demonstrates the best compression efficiency on all datasets, consistently outperforming both traditional codecs (H.265/HM, H.266/VTM) and previous state-of-the-art neural video codecs. This advantage is particularly pronounced in scenarios where bandwidth constraints or storage costs are critical, making our approach highly suitable for real-world applications.

In the higher quality range, the performance landscape becomes more heterogeneous. Our DCVC-UF (HT-L) achieves the best compression ratio on the HEVC E dataset, demonstrating its capability to handle specific content types effectively. However, on other datasets, VTM with Hierarchical-B configuration maintains its advantage in the very high quality range. This performance gap is expected, as it represents a deliberate design trade-off: our models prioritize computa-

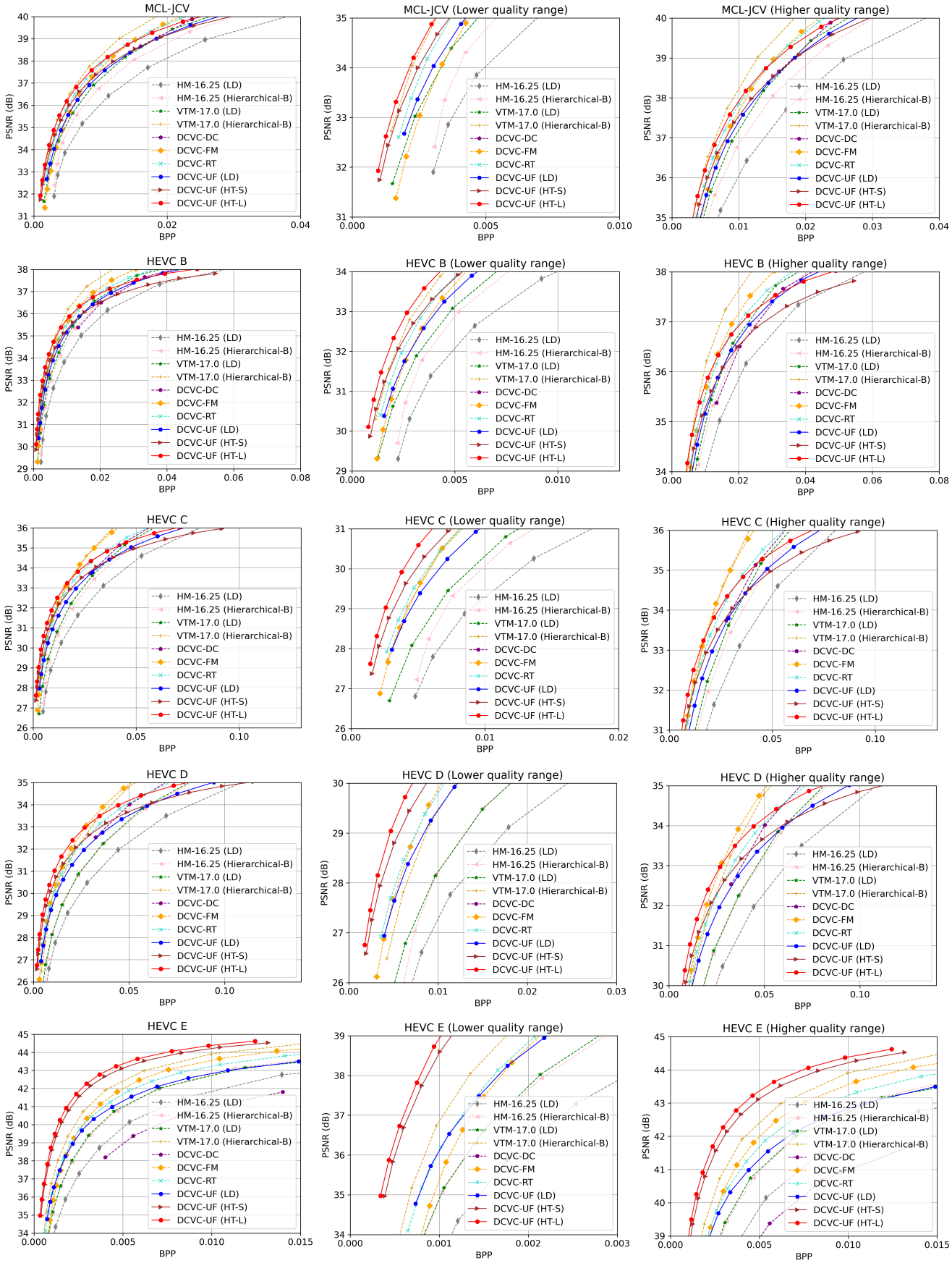

Figure 2. Rate-distortion curves.

tional efficiency while maintaining high compression ratio. The lightweight architecture that enables ultra-fast encoding and decoding speeds inherently limits the model capacity available for achieving maximum compression ratio at very high quality levels. These results underscore the practical applicability of our approach for scenarios requiring very fast processing with good compression ratio, while also identifying opportunities for future improvements in the high-quality domain through increased model capacity or more advanced design.

References

[1] HM. https://vcgit.hhi.fraunhofer.de/jvet/HM. 1

[2] VTM. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM. 1

[3] E. Alshina, J. Ascenso, T. Ebrahimi, F. Pereira, and T. Richter. [AHG 11] Brief information about JPEG AI CfP status. In *JVET-AA0047*, 2022. 1

[4] Anchors · JPEG-AI MMSP Challenge. Anchors · JPEG-AI MMSP Challenge. https://jpegai.github.io/7-anchors/. 1

[5] Zhaoyang Jia, Bin Li, Jiahao Li, Wenxuan Xie, Linfeng Qi, Houqiang Li, and Yan Lu. Towards practical real-time neural video compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-25, 2024*, 2025. 1, 2

[6] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22616–22626, 2023. 1