

Group-aware Parameter-efficient Updating for Content-Adaptive Neural Video Compression (Supplementary Materials)

ACM Reference Format:

. 2024. Group-aware Parameter-efficient Updating for Content-Adaptive Neural Video Compression (Supplementary Materials). In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (ACM MM' 24)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

This document is supplementary material for the paper “Group-aware Parameter-efficient Updating for Content-Adaptive Neural Video Compression”. We provide optimization details (Section A.1) adaptor details (Section A.2), more setting details (Section B.1), additional quantitative results (Section B.2), additional ablation studies (Section B.3) and individual quantitative results (Section B.4).

A METHODOLOGICAL DETAILS

A.1 Optimization Details

This section outlines our optimization strategy. Specifically, for the 1st GoP in the video, we utilize a pre-trained NVC model based on [19]. Initially, we optimize it over 1200 epochs with a learning rate of 1×10^{-5} . Throughout each epoch, the back-propagation operation is applied to all frames within the GoP. Subsequently, the learning rate is reduced to 5×10^{-6} , and the model is further optimized for an additional 600 epochs. For subsequent GoPs, we employ the parameters optimized from the previous GoP. The training for these GoPs starts with a learning rate of 1×10^{-5} for 600 epochs. Following this, the learning rate is decreased to 5×10^{-6} , and the training continues for another 300 epochs.

A.2 Adaptor Details

This section elaborates on the architectural decisions concerning the integration of adaptors into our content-adaptive NVC framework, initially introduced in Sec 3.2.1 of the main manuscript. As detailed in Fig. 2 (c) of the main manuscript, each adaptor consists of three convolutional layers. These include two point-wise convolution layers denoted as \mathbf{W}_{pre} and \mathbf{W}_{zero} , and one depth-wise convolution layer denoted as \mathbf{W}_{dw} . The subsequent discussion will detail the configuration and integration of these adaptor modules into various coding components of the NVC framework.

Serial Adaptor. In the light-weight coding components—namely, *Motion Estimation*, *Motion Hyper-Prior Encoder* and *Contexture Hyper-Prior Encoder*, a serial configuration is applied. This setup ensures

that the width and height of the input feature \mathbf{F} and the delta feature $\delta(\mathbf{F})$ remain identical. Consequently, the stride for all three convolutional operations is set to 1 to maintain dimension consistency. The depth-wise convolutional kernel \mathbf{W}_{dw} is utilized with dimensions of 3×3 . The output channel specifications for each component are as follows:

#Output Channel	\mathbf{W}_{pre}	\mathbf{W}_{dw}	\mathbf{W}_{zero}
Motion Estimation	32	32	2
Motion Hyper-Prior Encoder	16	16	64
Contexture Hyper-Prior Encoder	32	32	128

Table S1: Configuration Details of Serial Adaptors: The serial adaptors are integrated within various components of our NVC framework as detailed in Sections 3.1.2 and 3.2.1.

Parallel Adaptor. For those larger coding components, specifically the *Motion Encoder* and *Contexture Feature Encoder*, we implement parallel adaptors. Different from the serial configuration, the width and height of the input feature \mathbf{F} and the delta feature $\delta(\mathbf{F})$ do not need to match. This allows the stride of \mathbf{W}_{pre} to vary, enabling the reshaping of the feature resolution as required. However, we maintain a stride of 1 for both \mathbf{W}_{dw} and \mathbf{W}_{zero} , and keep the kernel size for \mathbf{W}_{dw} consistent at 3×3 . Further details and visual representations can be found in Fig. S1.

B EXPERIMENTS

B.1 More Setting Details

Settings of Traditional Codecs. We benchmark the performance of our method against the reference software for H.265/HEVC [17] and H.266/VVC [5], specifically using HM-16.20 [1] and VTM-11.2 [3], respectively. For both reference models, we employ the low delay configuration, which prioritizes the highest compression ratio. All video processing is conducted in YUV 420 format. Post compression, all RGB frames are extracted from the reconstructed videos to compute distortion metrics, allowing for a consistent evaluation of video quality across different coding standards. The detailed settings used for HM and VTM are as follows:

• HM

```
TAppEncoder -c encoder_lowdelay_main_rext.cfg [args]
```

• VTM

```
EncoderApp -c encoder_lowdelay_vtm.cfg [args]
```

where both codecs use the following common command line arguments ([args]):

```
--InputFile={input_filename}  
--BitstreamFile={bitstream_filename}  
--ReconFile={reconstructed_filename}  
--DecodingRefreshType=2
```

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or internal use, or for the internal or personal use of specific clients, is granted by ACM for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM MM' 24, 28 Oct–1 Nov, 2024, Melbourne, AU

© 2024 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

2024-04-20 07:42. Page 1 of 1–5.

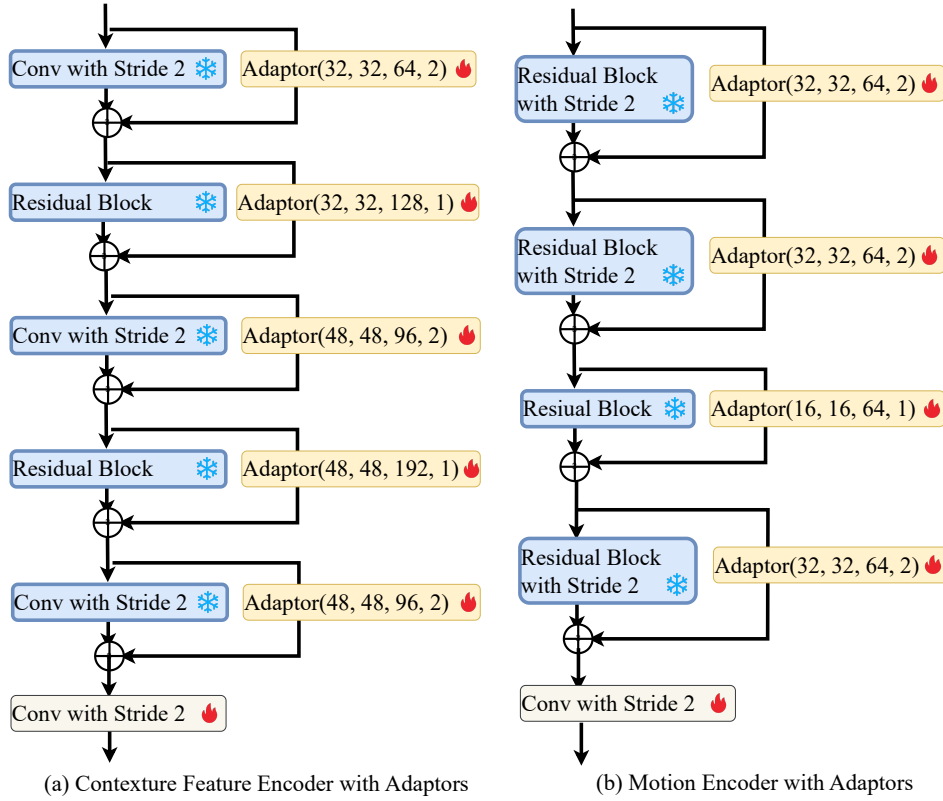


Figure S1: Configuration Details of Parallel Adaptors: The parallel adaptors are strategically integrated into specific coding components of our NVC framework. The configuration is denoted as $\text{Adaptor}(C_{pre}, C_{dw}, C_{zero}, S_{pre})$, where C_{pre} , C_{dw} , and C_{zero} represent the number of output channels in the convolutional operations of W_{pre} , W_{dw} , and W_{zero} , respectively. S_{pre} denotes the stride number of W_{pre} . Detailed descriptions are elaborated in Sec 3.1.2.

```
--InputBitDepth=8
--OutputBitDepth=8
--OutputBitDepthC=8
--InputChromaFormat=420
--FrameRate={frame_rate}
--FramesToBeEncoded={frame_num}
--SourceWidth={width}
--SourceHeight={height}
--IntraPeriod={GoP_Size}
--QP={quantization_parameter}
--Level=6.2
```

For more details regarding the size of GoP and frame number please refer to our Sec 4.1.2 of the main manuscript. For the resolution (i.e., width and height), please refer to the next section.

Settings of Patches. In this section, we outline the configuration for segmenting each GoP into patch-based GoPs for various video and medical sequences. Consistent with prior benchmarks, we first cropped the smaller dimension of all sequence frames as in [4, 6–10, 13–16]. Following this, we systematically partition each sequence from full resolution down to each sub-resolution. The specific segmentation details are as follows:

	Resolution	Patch Resolution	#Patch
HEVC B	1920 × 1024	320 × 256	24
HEVC C	832 × 448	416 × 224	4
HEVC D	384 × 192	384 × 192	1
HEVC E	1280 × 704	320 × 176	16
ACDC	256 × 224	256 × 224	1

Table S2: Configuration of Patch Settings for Each Dataset.

B.2 Additional Quantitative Comparison

Here, we present a more quantitative comparison between our method and other state-of-the-art video codecs using an additional public video dataset, the UVG dataset [2]. This dataset consists of 7 high-resolution videos (1920×1080). We processed and encoded the UVG dataset in the same manner as we did with the HEVC ClassB dataset. Further details can be found in Sec 4.1.2 and B.1.

Fig. S2 demonstrates the rate-distortion performance on the UVG dataset. As discussed in our Sec 4.2.3, given the smaller domain gap between the pre-trained dataset [19] and the test dataset, coupled with the nearing saturation of video compression benchmarks, it is

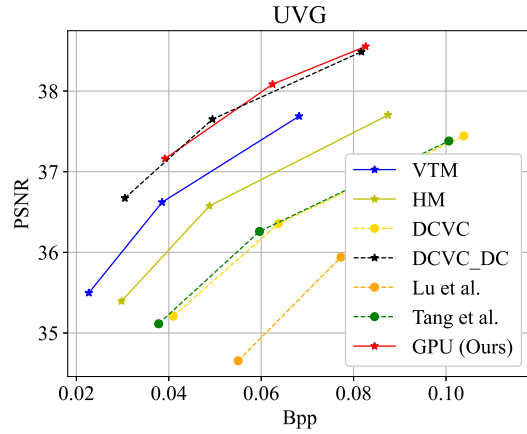


Figure S2: Rate-distortion (i.e., bit-rates vs PSNR) performance comparison of our and other state-of-the-art video codecs on additional video compression benchmark, UVG.

Table S3: The computational complexity on ClassD of our method using encoding-side adapters to delta-tune the encoder (i.e., GPU) against the variant directly optimizing all modules within the encoding network (i.e., Full Updating).

	Full Updating	GPU (Ours)
Optimizing Time (s/Epoch)	2.71	2.25

increasingly challenging to further enhance performance on such standard video compression benchmarks. Nevertheless, our method still outperforms the state-of-the-art NVC method DCVC_DC and the traditional video codec H.266/VVC, achieving bit-rate savings of 0.56% and 24.76%, respectively. Moreover, it shows significant performance gains over those content-adaptive NVC methods such as Lu *et al.* [12] and Tang *et al.* [18], saving 66.97% and 56.77% bit-rates. These results further underscore the effectiveness of our method in video compression

B.3 Additional Ablation Studies

In this section, we conduct experiments for the ablation study described in Sec 4.3 using the standard video dataset, HEVC ClassD [17]. We adhere to the experimental protocol described in Sec 4.3. Fig. S3 and S4 demonstrate the performance of our methods, comparing variants that update all encoder-side parameters (as detailed in Sec 4.3.1) and variants that involve different numbers of updated frames (as discussed in Sec 4.3.2), respectively.

The overall results are consistent with our main manuscript. In terms of optimizing efficiency, our GPU only incurs a negligible increase in bit-rates ($< 1\%$) compared to the full updating strategy, yet only requires 83.0% optimization time for each epoch (i.e., GoP) as shown in Table S3, and utilizes less than 10% of the parameters as shown in Table 2 of the manuscript. Regarding the impact of the number of frames optimized, optimizing 1, 10, and 20 frames results in additional bit-rate costs of 20.61%, 3.71%, and 1.27%, respectively.

2024-04-20 07:42. Page 3 of 1–5.

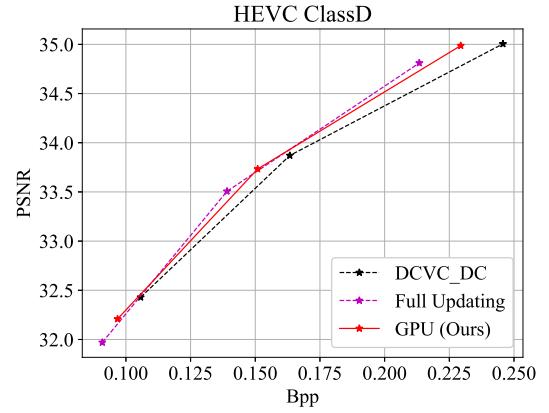


Figure S3: Performance of our method against the variant updating all encoder-side parameters (i.e., Full Updating) on the HEVC ClassD dataset.

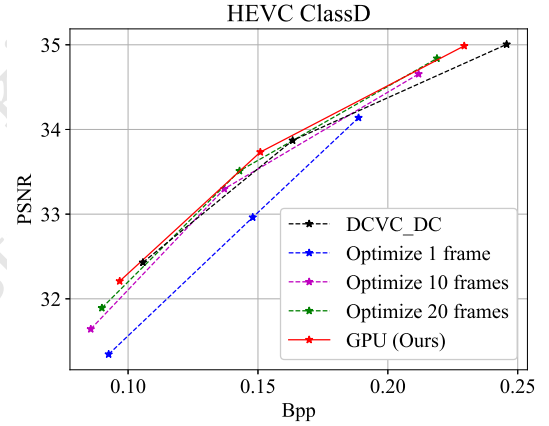


Figure S4: Performance of our method against variants optimizing different numbers of frames during online updating on the HEVC ClassD dataset.

B.4 Individual Quantitative Comparison

Lastly, we present selected rate-distortion performance curves for individual sequences from both the video and medical dataset. Specifically, we feature two cases for each dataset: the best (left column) and the worst (right column) cases. The best and worst cases are defined based on how our proposed content-adaptive NVC framework performs in comparison to our baseline method, DCVC_DC [11]. It is important to highlight that, on average, our NVC framework is able to outperform both the baseline DCVC_DC and the VTM methods across all datasets. This superiority is supported by the quantitative comparisons (please refer to the details in Fig. 4 and 5 and Table. 1 of the main manuscript).

Regarding the performance on the medical dataset, as shown in Fig. S6, our method significantly improves over the baseline in both the best and worst cases when compressing each MRI sequence. This further demonstrates its adaptability for individual cases.

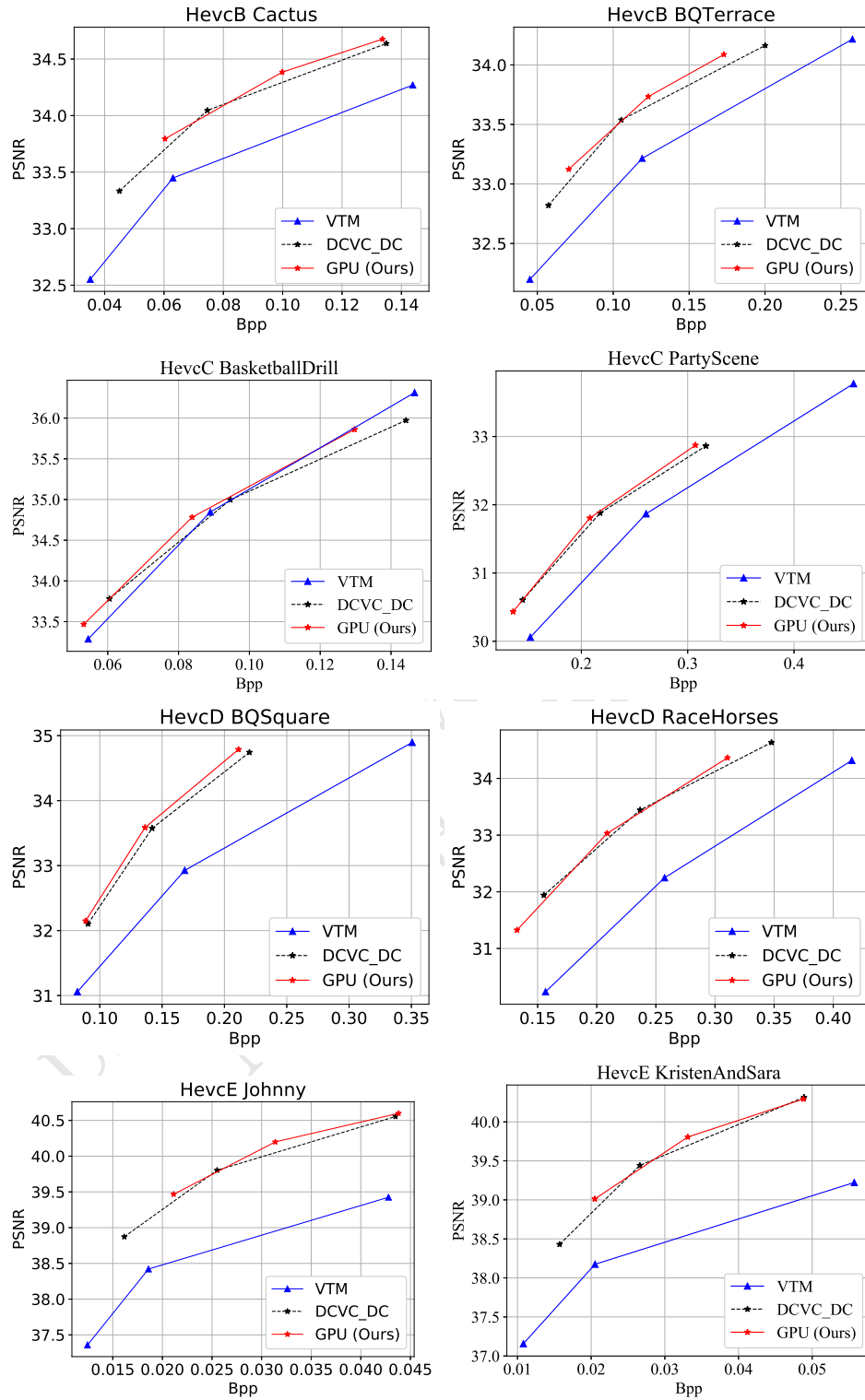


Figure S5: Selected individual rate-distortion (i.e., bit-rates vs PSNR) performance comparison on standard video dataset, HEVC B, C, D and E, showing best (left) and worst case (right) on the dataset.

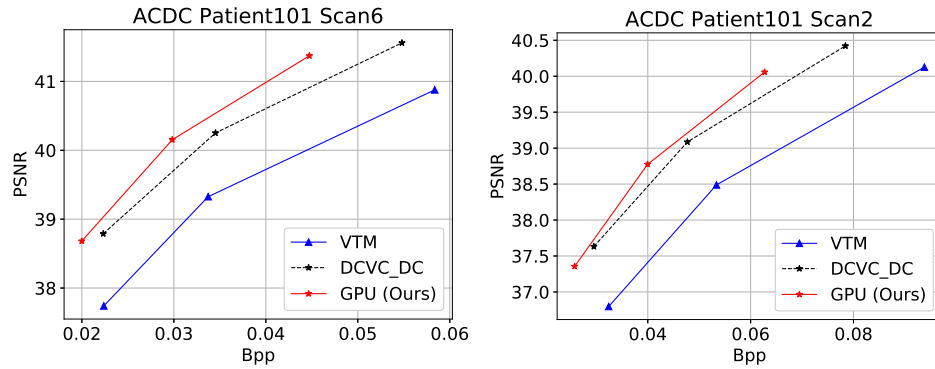


Figure S6: Selected individual rate-distortion (i.e., bit-rates vs PSNR) performance comparison of medical volumetric image dataset, ACDC, showing best (left) and worst case (right) on the datasets.

In terms of standard video compression, from Fig. S5, it can be observed that across the standard video benchmarks, HEVC Class B, C, D, and E, our proposed method consistently improves upon, or at least performs no worse than, the baseline DCVC_DC, even in the worst cases (e.g., HevcD RaceHorses). For cases where our method does not yield significant improvements, we plan to conduct detailed studies to explore the bottlenecks of such content-adaptive NVC methods. We believe these investigations will yield new insights for the video coding community.

REFERENCES

- [1] [n. d.]. Hvc test model (hm). <https://hevc.hhi.fraunhofer.de/HM-doc/>. Accessed: 2024-03-06.
- [2] [n. d.]. Ultra video group test sequences. <http://ultravideo.cs.tut.fi>. Accessed: 2023-03-06.
- [3] [n. d.]. VVC Reference Model (VTM). https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/. Accessed: 2024-03-06.
- [4] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. 2020. Scale-Space Flow for End-to-End Optimized Video Compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8503–8512.
- [5] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. 2021. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 10 (2021), 3736–3764.
- [6] Zhenghao Chen, Relic Lucas, Roberto Azevedo, Yang Zhang, Markus Gross, Dong Xu, Luping Zhou, and Christopher Schroers. 2023. Neural Video Compression with Spatio-Temporal Cross-Covariance Transformers. In *Proceedings of the 31th ACM International Conference on Multimedia*. ACM. <https://doi.org/10.1145/3581783.3611960>
- [7] Zhihao Hu, Zhenghao Chen, Dong Xu, Guo Lu, Wanli Ouyang, and Shuhang Gu. 2020. Improving deep video compression by resolution-adaptive flow coding. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II* 16. Springer, 193–209.
- [8] Zhihao Hu, Guo Lu, Jinyang Guo, Shan Liu, Wei Jiang, and Dong Xu. 2022. Coarse-to-fine Deep Video Coding with Hyperprior-guided Mode Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- [9] Zhihao Hu, Guo Lu, and Dong Xu. 2021. FVC: A New Framework towards Deep Video Compression in Feature Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1502–1511.
- [10] Jiahao Li, Bin Li, and Yan Lu. 2021. Deep contextual video compression. *Advances in Neural Information Processing Systems* 34 (2021), 18114–18125.
- [11] Jiahao Li, Bin Li, and Yan Lu. 2023. Neural Video Compression with Diverse Contexts. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, Canada, June 18–22, 2023*.
- [12] Guo Lu, Chunlei Cai, Xiaoyun Zhang, Li Chen, Wanli Ouyang, Dong Xu, and Zhiyong Gao. 2020. Content adaptive and error propagation aware deep video compression. In *European Conference on Computer Vision*. Springer, 456–472.
- [13] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. 2019. DVC: An end-to-end deep video compression framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11006–11015.
- [14] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. [n. d.]. An End-to-End Learning Framework for Video Compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* in Press ([n. d.]), 1–1. <https://doi.org/10.1109/TPAMI.2020.2988453>
- [15] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. 2020. An end-to-end learning framework for video compression. *IEEE transactions on pattern analysis and machine intelligence* 43, 10 (2020), 3292–3308.
- [16] Fabian Mentzer, George D Toderici, David Minnen, Sergi Caelles, Sung Jin Hwang, Mario Lucic, and Eirikur Agustsson. 2022. VCT: A Video Compression Transformer. *Advances in Neural Information Processing Systems* 35 (2022), 13091–13103.
- [17] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology* 22, 12 (2012), 1649–1668.
- [18] Chuanbo Tang, Xihua Sheng, Zhuoyuan Li, Haotian Zhang, Li Li, and Dong Liu. 2023. Offline and Online Optical Flow Enhancement for Deep Video Compression. *arXiv preprint arXiv:2307.05092* (2023).
- [19] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. 2019. Video enhancement with task-oriented flow. *International Journal of Computer Vision* 127, 8 (2019), 1106–1125.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009