An image to tailor: I-Frame Domain Adaptation in Neural Video Compression

Alberto. Presta
University of Turin, Italy
alberto.presta@unito.it

Gabriele Spadaro
University of Turin, Italy
LTCI, Télécom Paris
gabriele.spadaro@unito.it

Attilio Fiandrotti
University of Turin, Italy
LTCI, Télécom Paris
attilio.fiandrotti@unito.it

Marco Grangetto
University of Turin, Italy
marco.grangetto@unito.it

Abstract

Neural video compression (NVC) models recently outperformed traditional methods. They typically include an I-Frame codec for Intra-Frames and a P-Frame codec for P-frames. However, their performance may be far from optimal with data outside the training set. We propose domain adaptation (DA) in NVC using lightweight convolutional adapters inserted in the I-Frame decoder of a pre-trained NVC model, which are then fine-tuned. These adapters shift knowledge to a specific domain without altering the architecture or causing catastrophic forgetting. They enhance compression for both I-frames and P-frames while using minimal parameters with respect to the entire architecture, improving NVC robustness.

1 Introduction

Neural video compression (NVC) has attracted significant attention because it can outperform standardized codecs like H.266-VTM [3] in terms of compression efficiency. These models typically comprise two primary learnable networks, namely the I-frame and P-frame networks. The purpose of the I-frame network is to encode and decode the IntraFrames (I-Frames) independently form the other frames in the video. In contrast, the P-frame network uses previously reconstructed frames to learn contextual features that help minimize temporal redundancy. I-frames, though less frequent than P-frames, occupy more bitstreams due the fact that they cannot exploit any temporal context, so they are encoded as single images. Recent NVC advances focus on extracting a better spatial and temporal context to reduce redundancy [14, 23, 13, 15] in the P-frame model, leaving the latter almost unchanged. The framework uses backpropagation to minimize the rate-distortion (RD) loss $\mathcal{L} = \lambda\mathcal{D} + \mathcal{R}$, employing $\mathcal{D}$ for distortion, $\mathcal{R}$ for rate, and λ to balance the trade-off.

With hand-crafted modules replaced by learnable ones, NVC models heavily depend on their training datasets, limiting their adaptability to different contents. The limitation of handling off-domain content makes NVC models less attractive, jeopardizing their possible usage in a real scenario. A first naive method could involve fine-tuning the model using a dataset from a particular domain, thereby achieving optimal outcomes in that specific semantic content; however, this approach has two significant problems: finding the right hyperparameters for fine-tuning is complex, and there is a risk of catastrophic forgetting, where the network cannot solve the original task properly. A second approach involves the use of adapters [25, 5], small modules added to a pre-trained model and fine-tuned for a specific domain without altering the rest of the network. Although used in image compression domain adaptation [21, 26], they have not been exhaustively studied in NVC.

Accepted to the Neural Compression Workshop at the 38th Conference on Neural Information Processing Systems (NeurIPS 2024).

In this paper, we propose a hybrid approach in which we achieve domain adaptation in NVC through Convolutional Adapters (CADs) applied only at the decoder of the I-Frames model. The term hybrid means that we modify only the decoder of the I-frames model, leaving the P-frame one untouched and allowing P-frames encoding refinements based on I-frames. This approach offers several benefits. First, since we add adapters on the main architecture, we do not modify or replace any parameters, preventing catastrophic forgetting Second, focusing only on the decoder of I-frame model, we removed the rate term in the loss function, simplifying the optimization process.

Third, modifying only the I-Frames model improved coding efficiency in both I- and P- frames across various domain-specific datasets, underscoring the importance of a robust Intra model. This pioneering work in domain adaptation for NVC may inspire future advances.

2 Related work

Neural image compression (NIC) models use a hyperprior framework [2] to improve entropy parameter estimation by assuming a Gaussian distribution. Models employ autoregressive entropy estimation [19, 20] to capture spatial correlations and extract global and local contexts using attention modules, Transformers, or graphs with convolution [31, 32, 17, 24]..

Neural video compression (NVC) has also made significant improvements. Early works employed residual coding, leveraging the predicted, exploiting residuals to reduce redundancy. DVC [18] employs an optical flow network to produce the prediction frame, after which the residual between this frame and the current frame is encoded; This approach is improved by adding motion prediction [16, 22], scale-space optical flow estimation [1], and recurrent autoencoder [30]. Conditional coding [10, 11] implicitly learns contexts in the feature domain without being confined to the residuals of earlier frames in the pixel domain; in particular, the condition can be defined and learn, carrying richer information to improve efficiency. [10] introduced a single coder for both intra- and inter-frames using two subnets: the first transmits motion information and a coding mode that either performs actual coding with the second subnet or handles temporal prediction.

Majority of NVCs struggled to extract context due to their reliance only on optical flow for temporal correlation; in this sense a major effort has been made by the DCVC series [12, 13, 23, 14, 15], where higher dimension context is exploited. [12] initially explores methods to utilize, understand and establish a condition on NVC, introducing a high-dimensional feature domain context for the first time. [23] learned multiscale contexts from propagated features without relying on an auto-regressive entropy model, while [13] introduced a latent prior and dual spatial prior to efficiently capture temporal and spatial dependencies, reducing redundancy. [14] improved NVC by extracting diverse temporal and spatial contexts and using a quadtree-based partition for better entropy coding, while [15] improved quality by modulating latent features with a learnable quantizer and periodic refreshing for temporal feature regulation.

Domain adaptation (DA) has not been exhaustively studied in NIC and NVC. In [9], the generalized divisive normalization layers and the entropy model are fine-tuned with custom filters for the new dataset, requiring many parameters and challenging integration into a general architecture. [26] trained for each single image a tiny adapter, whose parameters must be sent with the bitstream, to perform single image optimization; despite the efficacy, this increases the time and computational complexity, jeopardizing its possible usage in NVC. [21] employed compact linear adapters, each dedicated to a specific target domain and utilizing 3×3 convolutions. Gating networks control these adapters for DA, boosting target domain performance and preventing catastrophic forgetting.

Most of the DA research on videos has focused on various topics, such as action recognition [28] or 3D segmentation [4]. However, NVC remains unexplored, prompting our research. Starting from the foundational DCVC models, this work seeks to tailor these codecs for specific domains, aiming to enhance the results even further.

3 Method

3.1 General architecture

Fig. 1 shows a blueprint of the NVC model. As mentioned in Sect. 2, it is formed by two submodels, namely the I-frame and the P-frame; the latter is depicted by the pink box in the figure.

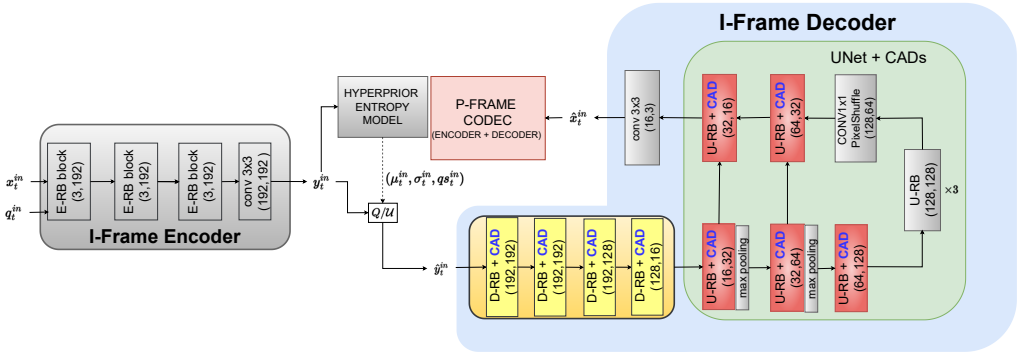
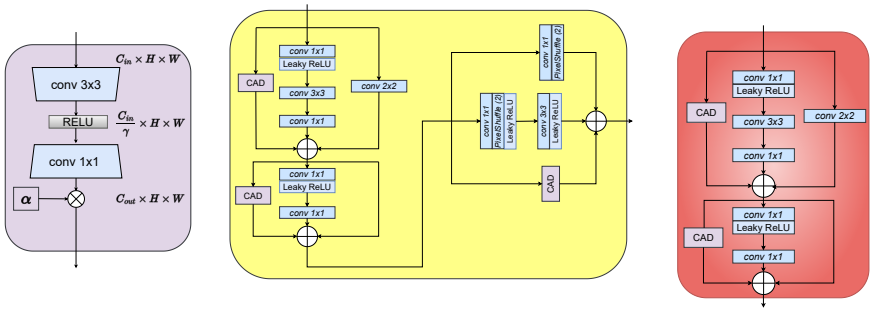


Figure 1: I-Frames network architecture. E-RB, D-RB, and U-RB denote encoder, decoder, and U-Net residual block respectively, while CAD stands for convolutional adapters, highlighted in dark-blue. Light-blue box denotes the entire I-frames Decoder.

P-Frame model learns and propagates temporal and spatial context to improve the encoding, decoding, and entropy modeling of the P-frames, exploiting both the I-frames and the information already collected. As this study intends to examine the impact of DA adapters within the I-frame model, we do not perform any modification of such architecture.

I-Frame model is designed to handle I-frames, which are frames encoded or decoded independently of any previous temporal context. Once reconstructed, these frames are utilized as input for the P-frame model to process additional frames.

We start from the model used in [14], which in turn is based on [27]. Denoting by x_t^{in} the intra frame at time t, it is projected to y_t^{in} with the I-Frame encoder, as shown in Fig. 1. y_t^{in} is then used as input for the hyperprior autoencoder to estimate the entropy parameters. Assuming the tensor follows a Gaussian distribution, μ and σ are determined to estimate the entropy during training and used for actual encoding during inference, utilizing the same entropy model as in [13]. The encoder also takes the desired quality q_t^{in} as input, enabling variable bitrate coding. This allows the hyperprior network to produce element-wise scales qs_t^{in} for latent vectors at different target qualities. Once the quantized tensor $\hat{y}_t^{in}$ is obtained, it is projected back into the pixel domain by the I-frame decoder (light-blue box), resulting in the reconstructed $\hat{x}_t^{in}$. After four decoding blocks where the number of channels is reduced from 192 to 16 (yellow box in figure), a final U-Net module (green box) and a convolutional layer are added to further improve the final reconstruction. Fig. 1 shows that we added convolutional adapters in the three final residual blocks of both the decoder (D-RB) and the U-Net (U-RB) and are represented by yellow and red boxes, respectively.



(a) CAD. (b) D-RB + CAD. (c) U-RB + CAD.

Figure 2: (a) Architecture of the Convolutional adapter (CAD), where C_{in} and C_{out} are the input and output channel dimension respectively, while H and W are the spatial dimensions. (b) Architecture of the D-RB with the addition of CADs. (c) Architecture of the U-RB with the addition of CADs.

3.2 Domain adaptation towards convolutional adapters

We inserted adapters only in the I-Frame decoder for two main reasons. First, by adding adapters for each domain, we avoid significantly increasing the number of parameters, in order not to overburden the model, keeping it manageable even with multiple domains. Furthermore, small adapters are easier to encode and transmit; for example, an adapter system with fewer than a million parameters can be sent in a 3.9 MB bitstream using 32-bit floats. Second, we simplified the resulting optimization problem as we do not need to address the rate minimization typically faced in NVC.

Convolutional adapters. Unlike [21], we implemented non-linear convolutional autoencoders (CAD) for each adapter, following [5]. Fig. 2a shows the structure of the adapters. When a tensor with C_{in} channels is input, it is projected into a latent space, reducing the number of channels to C_{in}/γ. In this context, γ is a hyperparameter: a larger γ results in a smaller latent space of the adapter, thus decreasing the number of required parameters, but also limiting its adaptability. We exploited 3×3 convolution in the CAD encoder since 1×1 convolution can result in suboptimal transferability due to different receptive fields [5], and a larger filter size in the CAD encoder preserves local sensitivity. We instead use 1×1 convolution in the CAD decoder to decrease parameters and maintain the same spatial dimension. Once the output with C_{out} channels is obtained, it is multiplied by the learnable parameters α regulating the importance of the adapters obtained.

CAD in the main architecture. Figures 2b and 2c show our integration of CADs within the residual blocks of both the deconvolutional decoders (D-RB) and U-Net (U-RB). Regarding the former (Fig. 2b), we integrated three CAD modules with identical values γ_d, one per residual sub-block, using a *residual parallel* strategy [5]. This setup enables CAD to fine-tune each component in parallel, balancing effectiveness and parameter efficiency. We follow the same strategy for the U-RB (Fig. 2c) where we inserted five CADs, with a γ_u independent of the one related to the D-RB blocks.

3.3 Training and inference

Given a fixed content-based domain D, we integrate CADs into selected positions of a pre-trained I-Frame codec and only fine-tune those specific parameters, leaving the remainder of the model unchanged. Since we only introduced adapters in the decoder of the I-frame model, the optimized loss function turns out to be only the distortion between the original I-frame and its reconstruction, denoted by $\mathcal{L} = \mathcal{D}(x_t^{in}, \hat{x}_t^{in})$. In our case, we optimize the CADs for mse and $ssim$ separately, as maintaining this separation is a common practice in both NIC and NVC. Once we complete the fine-tuning of the adapters, we used the obtained network as I-Frame codec in a pre-trained NVC; to perform actual entropy coding, we exploit an ANS codec [6].

4 Experiments

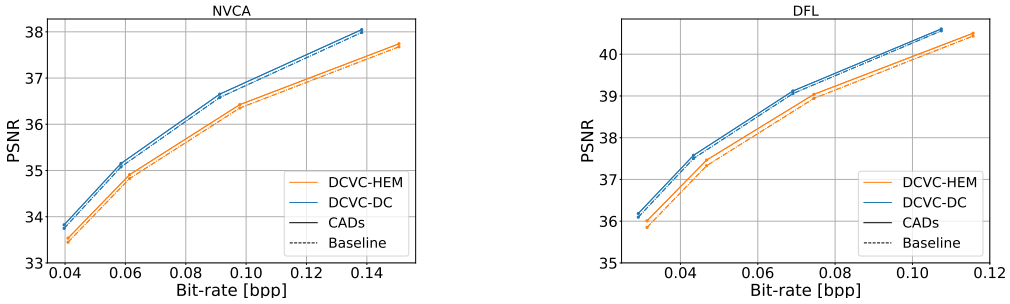
4.1 Training setup

To validate our methods, we used two domain-specific datasets. NSVA [29] with high-quality NBA (*national basketball association*) basketball videos, and DFL [7] with high-quality German soccer league videos (*Bundesliga*). We selected these datasets because they are high-resolution, feature long sequences (hundreds of frames), and represent realistic scenarios for real-time streaming services.

We set (γ_u,γ_d) to (4,8), assigning more parameters and greater transferability to the decoder rather than the U-Net, demonstrating its potential for improved outcomes. We trained our adapters for 50 epochs, using Adam as an optimizer with an initial learning rate of 10^{-4}, halving it when reaching a plateau with patience in 5 epochs. We tested our methods on two NVC models, namely DCVC-HEM[13] and DCVC-DC[14], which are part of the SOTAs on this topic, and we consider *PSNR* vs. Bits per pixel (bpp) as standard metrics. We encode an I-frame for every 32 P-frames, that is, we fixed a group of pictures (GOP) of 32. We used both DCVC-DC and DCVC-HEM I-frame models; the latter closely follows the one in sect 3, except for some channel dimensions.

4.2 RD performance

Figure 7 shows the RD performance for all frames (both I and P) in terms of PSNR considering 15 test videos from NVCA (a) and DFL (b), using DCVC-HEM [13], and DCVC-DC [14] as the starting baseline models; to be fair in comparison, we maintain the same I-Frame codec across



(a) (b)

Figure 3: RD plot (PSNR vs. bpp) for the NVCA (a) and DFL (b) datasets, comparing CAD utilization with DCVC-HEM, and DCVC-DC. Dotted lines indicate reference results, while the continuous lines show the results obtained using our method. The MS-SSIM results are inserted in the appendix.

these different NVC models, with and without CADs. As observed from the continuous lines, we successfully surpassed the baseline results in both datasets and all the two foundational models evaluated. Although the improvements may appear marginal in the figures, they were achieved by modifying only a small portion of the I-frame codec, which encodes 1 frame out of every 32. Consequently, most of the architecture remains unchanged. Table 1 presents the Bjontegaard metrics [8], using DCVC-* models as baselines; we analyzed three different types of frames, namely intra (I), predicted (P) and all frames. Even with a slight increase in additional parameters, specifically less than 2% of the entire NVC network, we observe an improvement in overall performance in the target datasets. For example, in the NVCA dataset, BD-RATE improves by more than 2%. It is important to note that we achieved this simply by adding adapters without altering the original parameters, thus avoiding the issue of catastrophic forgetting. This approach is crucial to preserve robustness and not jeopardize the usability of such models. It is important to note that the enhancements observed in the results are attributed not solely to the improvements in the I frames, but also to the P frames, as evidenced by the negative BD-RATE values in Tab. 1. This indicates that an optimal I-frame codec can also enhance the efficiency of P-frames. Moreover, our results were achieved by adding a minimal number of additional parameters, as demonstrated by the final columns of the tab. 1; for instance, for DCVC-DC, we incorporated 948k parameters, which is less than 2% of the entire NVC model. This is vital for practical scenarios needing storage of diverse adapters for different domains.

Table 1: Comparison of BD-RATE/PSNR and parameters between our proposed method and state-of-the-art NVC models. For BD-* calculations, we used DCVC-* as the reference models. *Add. pars (%)* represents the additional parameters in percentage added to the main architecture.

	All		***I***		***P***		***# parameters***	
	BD-RATE	BD-PSNR	BD-RATE	BD-PSNR	BD-RATE	BD-PSNR	CAD	Add. Pars (%)
	NVCA							
DC + CAD	-2.10	0.07	-7.45	0.41	-1.85	0.05	948k	1.85
HEM + CAD	-2.38	0.08	-8.14	0.44	-2.16	0.066	956k	1.96
DFL								
DC + CAD	-1.98	0.068	-3.60	0.22	-1.94	0.007	948k	1.85
HEM + CAD	-3.22	0.11	-8.64	0.48	-1,93	0.06	956k	1.96

4.3 Varying the number of I-frames

We analyzed CADs compression enhancement by varying the GOP. Figure 4a shows how BD-RATE changes with respect to the GOP, considering DCVC-DC as the base model, and on NVCA dataset; we also point out that in this experiment we encode only the first 128 frames. As evident, for all frames (orange) and when considering just P frames (blue), we consistently achieve significant improvements. Even in the extreme case where only the first frame is encoded as I (GOP = 128), we achieve a BD-RATE of -0.5 dB. Since we only modify the I-frame model, as the GOP increases, the performance gap between our method and baseline models decreases due to the increased distance between improved I-frames. This is shown in Fig. 4b with a sample video from the NVCA dataset, where we plot the Δ-PSNR (frame-by-frame PSNR difference between reconstructions with and

without CADs). Moving from GOPs of 32 to GPs of 64 pictures, the difference gradually decreases until the next I frame is reached. However, it is clear that even if we add a negligible number of parameters, we managed to obtain remarkable improvements, especially in such situations where it is necessary to maintain a small value of GOP.

(a) (b)

Figure 4: (a) BD-RATE for various GOP settings on the NVCA dataset using DCVC-DC + CADs, with DCVC as the baseline model. Here we encoder the first 128 frames for each sample. (b) Δ-PSNR on a video sample from NVCA dataset, considering a GOP of 32 and 64.

4.4 Ablation study

Tab 2 presents BD-RATE, BD-PSNR, trainable parameters, and the percentage of additional parameters (Add. Pars) for different configurations. We modify γ_d and γ_u, and compared the CADs with fine-tuning of the decoder. we exploited DCVC-DC as the base model and the NVCA dataset.

BD-RATE indicates that D-RBs have more adaptation ability than U-RB (BD-RATE of -2.10 versus -1.99); furthermore, reducing both γ_d and γ_u decreases parameters, but also performance improvements. Finding the best configurations means balancing these two factors and it depends on the practical scenario; for example, a configuration with $(\gamma_d, \gamma_u) = (4, 8)$ produces a BD-RATE of -2.1 with training parameters under 2% compared to the base model. Alternatively, fine-tuning the entire decoder improves performance but leads to catastrophic forgetting by modifying the original parameters and increases the parameters to nearly 4.5M, making it less efficient than using CADs.

Table 2: Comparison of CADs considering different configurations, different type of adapters, and the fine-tuning technique. In bold we highlight the default configuration used in Sect. 4.2.

type of adapters	(γ_d, γ_u)	BD-RATE	BD-PSNR	# pars	Add Pars. (%)
CADs	**(4,8)**	**-2.10**	**0.07**	**948k**	**1.85**
CADs	(8,4)	-1.99	0.068	572K	1.12
CADs	(8,8)	-1.91	0.064	508k	0.99
CADs	(6,6)	-1.96	0.067	667k	1.3
CADs	(0,8)	-1.37	0.047	65K	0.12
CADs	(4,0)	-1.92	0.065	883K	1.73
Dec + Unet Refine	-	-2.99	0.1	4.497M	8.81

5 Conclusion

We incorporate convolutional adapters (CADs) into the I-frame decoder for content-based domain adaptation in NVC. Such models were incorporated into the decoder and U-net architecture and then fine-tuned for domain-specific reconstruction. These CADs were used solely in the I-frame model, leaving the rest of the NVC unchanged. The tests in two datasets showed significant compression improvements with less than 2% additional parameters, enhancing both the I-frame and P-frame reconstructions; such results suggest that an efficient I-frame codec positively impacts the entire NVC system, especially with a smaller GOP. This work explores content-based DA in NVC, a largely unexplored area. Few CADs in the I-frames decoder yield promising results, suggesting improvements by adding adapters in the P-frames encoding process and extending our experiments to more NVC models.

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*, pages 8503–8512, 2020.

[2] Johannes Ballé, , et al. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

[3] 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 Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021.

[4] Haozhi Cao, Yuecong Xu, Jianfei Yang, Pengyu Yin, Shenghai Yuan, and Lihua Xie. Multi-modal continual test-time adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18809–18819, 2023.

[5] Hao Chen, Ran Tao, Han Zhang, Yidong Wang, Xiang Li, Wei Ye, Jindong Wang, Guosheng Hu, and Marios Savvides. Conv-adapter: Exploring parameter efficient transfer learning for convnets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1551–1561, 2024.

[6] Jarek Duda. Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding. *arXiv preprint arXiv:1311.2540*, 2013.

[7] Jakub Michalczyk et al. Dfl - bundesliga data shootout, 2022.

[8] Bjontegaard Gisle. Calculation of average psnr differences between rd curves. In *ITU-T SG16/Q6, 13ˆ< th> VCEG Meeting, Austin, Texas, USA, April 2001*, 2001.

[9] Sudeep Katakol, Luis Herranz, Fei Yang, and Marta Mrak. Danice: Domain adaptation without forgetting in neural image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1925, 2021.

[10] Théo Ladune, Pierrick Philippe, Wassim Hamidouche, Lu Zhang, and Olivier Déforges. Conditional coding for flexible learned video compression. *arXiv preprint arXiv:2104.07930*, 2021.

[11] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34:18114–18125, 2021.

[12] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34:18114–18125, 2021.

[13] Jiahao Li, Bin Li, and Yan Lu. Hybrid spatial-temporal entropy modelling for neural video compression. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.

[14] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, Canada, June 18-22, 2023*, 2023.

[15] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with feature modulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 17-21, 2024*, 2024.

[16] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. M-lvc: Multiple frames prediction for learned video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3546–3554, 2020.

[17] Jinming Liu et al. Learned image compression with mixed transformer-CNN architectures. In *CVPR*, 2023.

[18] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11006–11015, 2019.

[19] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. *NeurIPS*, 31, 2018.

[20] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *ICIP*, 2020.

[21] Alberto Presta, Gabriele Spadaro, Enzo Tartaglione, Attilio Fiandrotti, and Marco Grangetto. Domain adaptation for learned image compression with supervised adapters. In *2024 Data Compression Conference (DCC)*, pages 33–42. IEEE, 2024.

[22] Oren Rippel, Alexander G Anderson, Kedar Tatwawadi, Sanjay Nair, Craig Lytle, and Lubomir Bourdev. Elf-vc: Efficient learned flexible-rate video coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14479–14488, 2021.

[23] Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu. Temporal context mining for learned video compression. *IEEE Transactions on Multimedia*, 2022.

[24] Gabriele Spadaro, Alberto Presta, Enzo Tartaglione, Jhony H. Giraldo, Marco Grangetto, and Attilio Fiandrotti. Gabic: Graph-based attention block for image compression. In *2024 IEEE International Conference on Image Processing (ICIP)*, 2024.

[25] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5227–5237, 2022.

[26] Koki Tsubota, Hiroaki Akutsu, and Kiyoharu Aizawa. Universal deep image compression via content-adaptive optimization with adapters. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2529–2538, 2023.

[27] Guo-Hua Wang, Jiahao Li, Bin Li, and Yan Lu. Evc: Towards real-time neural image compression with mask decay. In *International Conference on Learning Representations*, 2023.

[28] Pengfei Wei, Lingdong Kong, Xinghua Qu, Yi Ren, Zhiqiang Xu, Jing Jiang, and Xiang Yin. Unsupervised video domain adaptation for action recognition: A disentanglement perspective. *Advances in Neural Information Processing Systems*, 36:17623–17642, 2023.

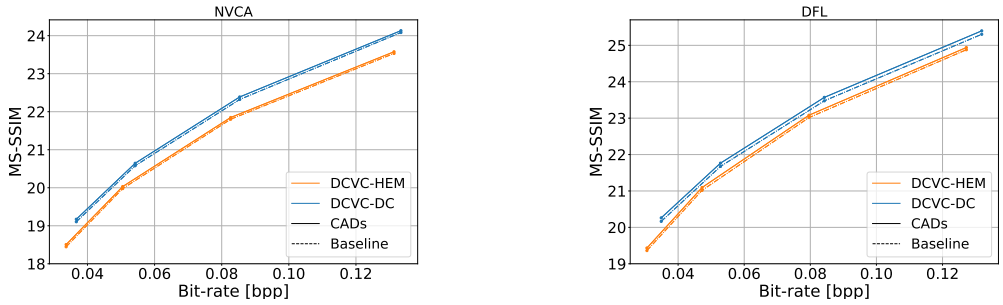
[29] Dekun Wu, He Zhao, Xingce Bao, and Richard P. Wildes. Sports video analysis on large-scale data. In *ECCV*, Oct. 2022.

[30] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[31] Lei Zhou et al. End-to-end optimized image compression with attention mechanism. In *CVPRW*, 2019.

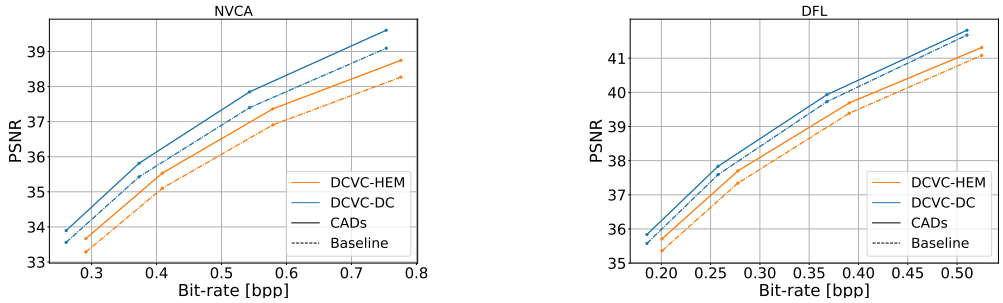
[32] Renjie Zou, Chunfeng Song, and Zhaoxiang Zhang. The devil is in the details: Window-based attention for image compression. In *CVPR*, 2022.

A Supplemental material



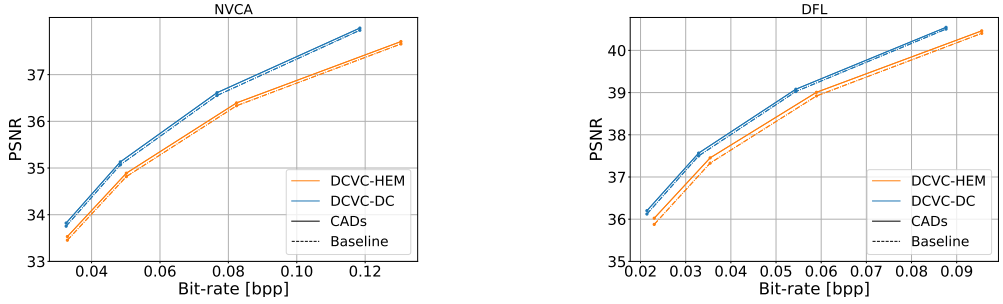
(a) (b)

Figure 5: RD plot (MS-SSIM vs. bpp) for the NVCA (a) and DFL (b) datasets, comparing CAD utilization with DCVC-HEM and DCVC-DC. Dotted lines indicate reference results, while the continuous lines show the results obtained using our method.



(a) (b)

Figure 6: RD plot (PSNR vs. bpp) for the NVCA (a) and DFL (b) datasets, comparing CAD utilization with DCVC-HEM and DCVC-DC considering only I-Frames. Dotted lines indicate reference results, while the continuous lines show the results obtained using our method.



(a) (b)

Figure 7: RD plot (PSNR vs. bpp) for the NVCA (a) and DFL (b) datasets, comparing CAD utilization with DCVC-HEM, and DCVC-DC considering only P-Frames. Dotted lines indicate reference results, while the continuous lines show the results obtained using our method.