CONTENT-ADAPTIVE MOTION RATE ADAPTION FOR LEARNED VIDEO COMPRESSION

Chih-Hsuan Lin Yi-Hsin Chen Wen-Hsiao Peng
{meerkat10.cs09g@, yhchen.iie07g@, wpeng@cs.}nctu.edu.tw
Computer Science Dept., National Yang Ming Chiao Tung University, Taiwan

***Abstract*—This paper introduces an online motion rate adaptation scheme for learned video compression, with the aim of achieving content-adaptive coding on individual test sequences to mitigate the domain gap between training and test data. It features a patch-level bit allocation map, termed the α-*map*, to trade off between the bit rates for motion and inter-frame coding in a spatially-adaptive manner. We optimize the α-*map* through an online back-propagation scheme at inference time. Moreover, we incorporate a look-ahead mechanism to consider its impact on future frames. Extensive experimental results confirm that the proposed scheme, when integrated into a conditional learned video codec, is able to adapt motion bit rate effectively, showing much improved rate-distortion performance particularly on test sequences with complicated motion characteristics.**

***Index Terms*—content-adaptive learned video compression, conditional inter-frame coding, bit allocation.**

I. Introduction

End-to-end learned video compression has been an active research area since the advent of the seminal work DVC [1]. Much research has been focused on improving temporal prediction for residual coding in the pixel domain [2]–[5] or the feature domain [6]. Recently, a new school of thoughts emerged, replacing residual coding with conditional coding [7]–[9] and making a significant breakthrough in compression performance.

Although showing promising coding results, learned video codecs may suffer from generalization issues. The domain gap between training and test data often leads to their sub-optimal coding performance and/or poor generation on individual test sequences.

With the aim of achieving better rate-distortion performance on individual test sequences at inference time, content-adaptive coding with learned codecs attracts lots of attention. [10]–[12] propose to optimize the latent representations of individual images/sequences or the encoder through back-propagating a rate-distortion loss at inference time. [13], [14] further include the decoder for end-to-end optimization and signal the optimal decoder parameters in the bitstream. Taking a different approach, [15] reserves external coding options for learned codecs to perform resolution-adaptive coding of the flow map or the image latents. Similarly, [16] creates a pixel-wise importance map that can be specified for spatially-adaptive image coding. While the aforementioned methods may invoke back-propagation or rate-distortion optimization at inference time, [17] represents a feed-forward approach that uses a neural network to predict coding modes from observing the hyperprior for resolution-adaptive flow and residual coding.

This work was supported by the Higher Education Sprout Project of the National Yang Ming Chiao Tung University, Ministry of Education (MOE), Taiwan, MOST, Taiwan, under Grant MOST 110-2221-E-A49-065-MY3 and National Center for High-performance Computing.

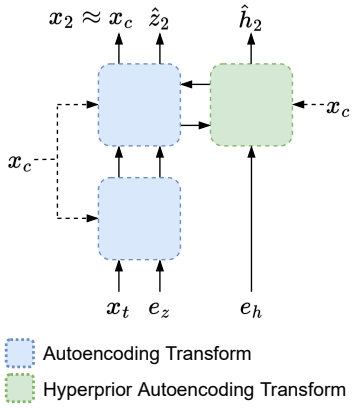


Fig. 1: Conditional augmented normalizing flow-based inter-frame codec [9].

Inspired by [16], we introduce a patch-level bit allocation map, termed the α-*map*, to a learned video codec. The α-*map* offers a mechanism to trade off between the bit rates for motion and inter-frame coding in a spatially-adaptive manner. To optimize the α-*map* for content-adaptive coding, we propose an online back-propagation scheme with look-ahead to consider its impact on future frames. Extensive experimental results confirm the effectiveness of the proposed method.

II. Related Work

We base our scheme on a conditional augmented normalizing flow-based (CANF-based) inter-frame codec [9]. CANF encodes an input frame x_t conditioned on its motion-compensated reference frame x_c. Fig. 1 depicts the framework of CANF-based inter-frame codec. The encoding process transforms the augmented inputs (x_t, e_z, e_h) into the latent representations $(x_2, \hat{z}_2, \hat{h}_2)$ by conditional autoencoding and hyperprior transforms. The latent variables $\hat{z}_2$ and $\hat{h}_2$ captures

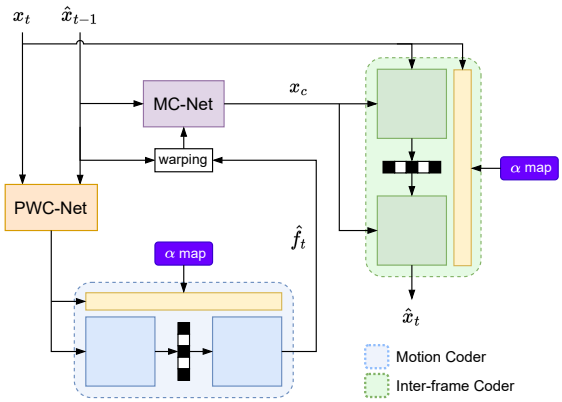


Fig. 2: The P-frame coding architecture with our proposed scheme.

the information needed to signal the transformation from the input x_t to x_2, which is regularized to approximate x_c during training. The decoding process first sets x_2 to x_c, followed by decoding $\hat{z}_2$ and $\hat{h}_2$ to perform inverse transformation from x_c to x_t.

III. Proposed Method

A. System Overview

Fig. 2 depicts the P-frame coding architecture with our proposed scheme. It comprises a flow estimation network (PWC-Net [18]), a motion compensation network (MC-Net), and two α-*map*-guided codecs, which are the motion codec and the conditional inter-frame codec. The former encodes the optical flow map estimated between the coding frame x_t and its reference frame $\hat{x}_{t-1}$, while the latter is adapted from [9] to encode the coding frame x_t conditionally on the motion-compensated reference frame x_c. x_t and $\hat{x}_{t-1}$ are of size $W \times H$. In this work, the α-*map* of dimension $W/64 \times H/64$ serves as a prior conditioning signal used to trade off between the bit rates consumed by the motion and the inter-frame codecs. Each component $\alpha_i \in [-1, 1]$ in the α-*map* is a real number that corresponds to a distinct 64×64 patch i in the input frame. By altering the α-*map*, a spatially-varying trade-off between the bit rates for motion coding and inter-frame coding is achieved. Moreover, the α-*map* is adapted on a frame-by-frame basis, allowing frame-adaptive optimization.

B. Conditional Feature Transformation with The α-Map

To adapt the P-frame coding pipeline to the α-*map*, we incorporate Spatial Feature Transform (SFT) layers and SFT Residual Blocks (SFT Resblk) [16], [19] into the motion and the conditional inter-frame codecs. Take our CANF-based inter-frame codec in Fig. 3 as an example. SFT applies spatially-adaptive affine transformation to the latent features F in the encoding/decoding transforms, with the element-wise affine parameters (γ, β) derived from the prior conditioning modules. In other words, $SFT(F|\gamma, \beta) = \gamma \odot F + \beta$. It is to be noted that the inputs to the prior conditioning modules include not only the α-*map*, but also the corresponding input signals to the encoding/decoding transforms. Before being concatenated with these input signals, the α-*map* (of size $W/64 \times H/64$) is scaled accordingly to match their dimensions.

During the encoding process, the target x_t is transformed into x_c, where the latent $\hat{z}_2$ captures the information needed to signal the transformation while the α-*map* determined externally (Section III-D) is fed to the prior conditioning modules to adapt the latent features. In particular, we signal the α-*map* implicitly in the hyperprior $\hat{h}_2$. That is, during decoding when x_t is recovered from x_c, an approximate α-*map* is extracted from $\hat{h}_2$ by a lightweight network. Notably, the discrepancy in α-*map* for encoding and decoding may contribute to the reconstruction error of x_t.

Our motion codec follows a similar architecture to [20] and is likewise guided by the α-*map*.

C. Training Objective

We adopt the following objective function to train our system end-to-end. The patch-level bit rate R_{M_i} for motion coding is weighted exponentially with a factor δ^{α_i} against the patch-level bit rate R_{R_i} for inter-frame coding according to the α-map.

$$L = \lambda \times D + R_W, \tag{1}$$

$$R_W = \sum_{i=1}^{N} \delta^{\alpha_i} \times R_{M_i} + R_{R_i}, \tag{2}$$

where the base $\delta = 10$ of the exponential is chosen empirically to compensate for the uneven ratio between R_{M_i} and R_{R_i}. N is the number of 64×64 patches in the input frame. It is seen that the model is trained to suppress R_{M_i} for higher R_{R_i} when $\alpha_i = 1$ and otherwise when $\alpha_i = -1$. R_{M_i}, R_{R_i} are weighted equally by setting $\alpha_i = 0$. During training, the α-*map* is randomized by having $\alpha = tanh(x)$, where x is drawn from a standard normal distribution. This ensures that the model is able to react to any α-*map* given at inference time.

D. Determining The α-Map

We determine the α-*map* for content-adaptive bit allocation between motion and inter-frame coding. To this end, we propose an online back-propagation scheme. The idea is to consider the α-*map* associated with each input frame as coding parameters to be updated on-the-fly by back-propagation. We take the pre-trained model from Section III-C and minimize Eq. (1) with respect to the α-*map*, with R_W taking the form of $\sum_i^N R_{M_i} + R_{R_i}$, where we discard the factor δ^{α_i} because we wish to arrive at an α-*map* that can best trade off between the bit rates for motion and inter-frame coding in order to minimize the rate-distortion cost for the current coding frame. In a sense, this approach is sub-optimal because it optimizes greedily the α-*map* of a coding frame without regard to its impacts on future frames (see Fig. 4a).

To explore the potential of our scheme, we additionally experiment with a look-ahead mechanism that optimizes the α-*map* of a coding frame by taking into account its impact on future frames. The idea is illustrated in Fig. 4b, where we minimize the sum of the rate-distortion costs over two

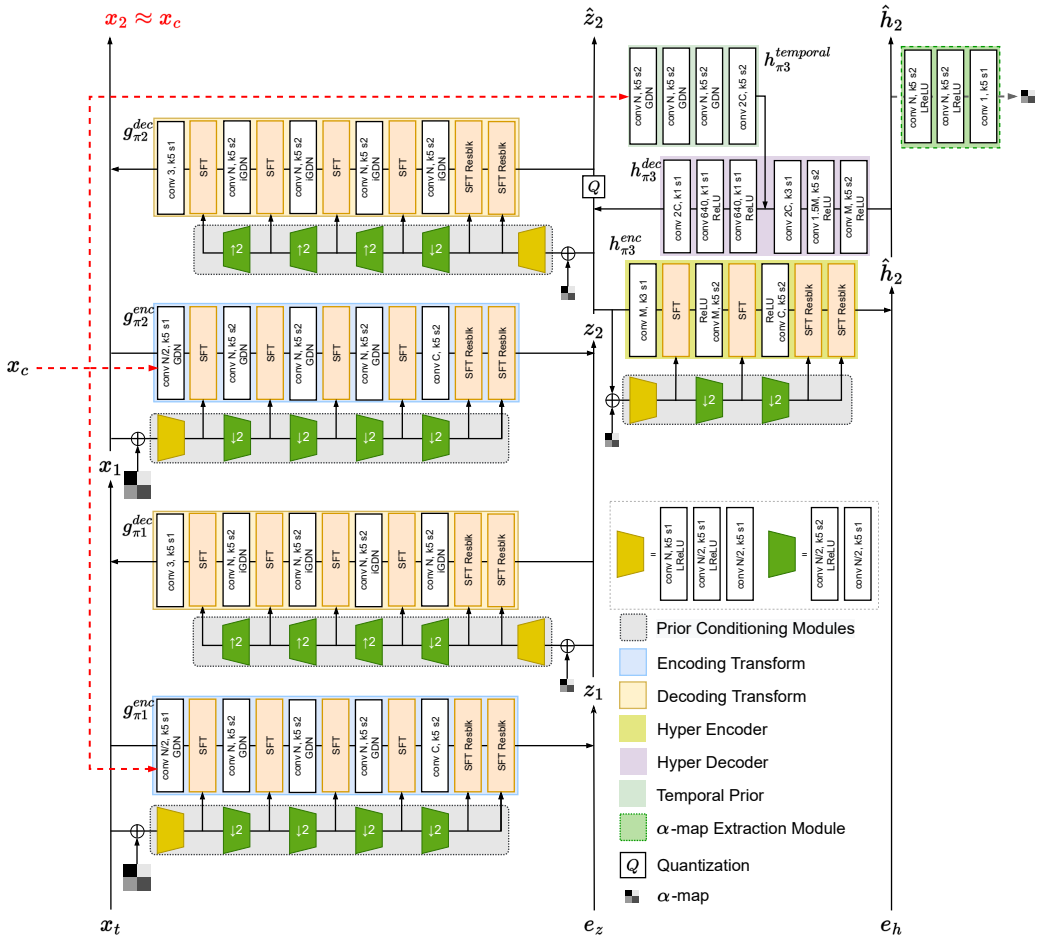
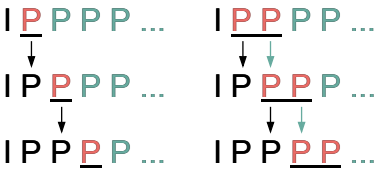


Fig. 3: The detailed architecture of our α-*map*-guided conditional inter-frame codec, where N=C=128, M=192. Following [9], we fix e_z at 0 during training and evaluation. For the hyperprior branch, we draw $e_h \sim \mathcal{U}(-0.5, 0.5)$ for simulating the quantization of the hyperprior $\hat{h}_2$ during training, and set it to 0 when $\hat{h}_2$ is rounded during evaluation.



(a) (b)

Fig. 4: The determination of the α *map* (a) without and (b) with the look-ahead mechanism. Black, red, and gray colors refer to coded frames, coding frames, and frames to be coded, respectively.

consecutive video frames by updating their α-*maps* simultaneously. In particular, the resulting α-*map* of the first frame in display order is used for coding the first frame, whereas that of the second frame serves as its initial α-*map*, which is to be further optimized together with the subsequent frame in a sliding window manner.

IV. Experiments

A. Settings

Training details: We use Vimeo-90k dataset [21] to train our model in two stages. First, we train our model without the prior conditioning modules. We then include the prior conditioning modules to train the entire system end-to-end. The Vimeo-90k dataset contains 91,701 sequences of size 448×256. Video frames are randomly cropped to 256×256 for training. We adopt Adam optimizer [22]. The learning rate is fixed at $1e^{-4}$ before 300k iterations, and is then decreased to $1e^{-5}$. The λ in Eq. (1) is set to 256, 512, 1024, 2048.

Evaluation methodologies: For evaluation, we test our method on UVG [23], HEVC Class D [24] and 5 challenging sequences from CLIC'22 [25] validation and test datasets. The sequences selected from CLIC'22 have complex motion, e.g. fast motion, zoom-in, or rotation. Remarkably, we downscale the sequences in UVG (of size 1920×1080) and CLIC'22 (of size 1920×1080 or 2048×1080) by a factor of 4 (denoted as UVG* and CLIC-MIX* in Table II), in order to facilitate the α-*map* optimization with limited GPU memory. We set the intra period to 32, and encode the first 96 frames for all the test sequences. We evaluate the reconstruction quality in PSNR-RGB and the bit-rate in bits-per-pixel (bpp).

Baseline methods: The anchor for comparison uses the

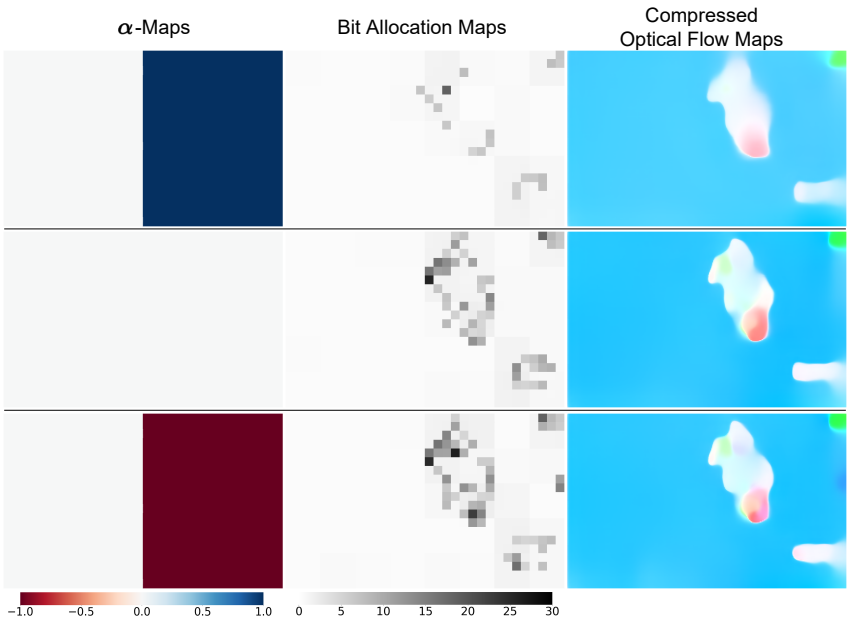


Fig. 5: Visualization of the α-*maps*, the bit allocation maps for motion coding, and the compressed optical flow maps.

TABLE I: Changes in bit rate for motion and inter-frame coding in response to varying the α-*map*.

	Bit Rate (10^{-3} bpp)	
	Motion Coding	Inter-frame Coding
α=1	1.67 (-24.8%)	51.85 (+6.5%)
α=0	2.22 (0.0%)	48.67 (0.0%)
α=-1	2.86 (+28.8%)	47.13 (-3.2%)

same learned video codec as ours. Specifically, it adopts ANFIC [20] for I-frame coding and the architecture in Fig. 2 for P-frame coding. Particularly, it sets the α-*map* uniformly to 0; in other words, there is no content-specific optimization for motion and inter-frame coding. We additionally compare our scheme with DCVC [8], which is the state-of-the-art learned video codec. For a fair comparison, we also use ANFIC [20] as the I-frame codec for DCVC [8].

B. Effectiveness of The α-map

Table I shows the trade-off between the bit rates used for motion and inter-frame coding on UVG dataset when we alter the α-*map* from 1 to -1. In this experiment, the α-*map* has a uniform value across spatial locations. It is seen that as compared to $\alpha = 0$, the bit rate for motion coding decreases by nearly 25% (respectively, increases by nearly 29%) when $\alpha = 1$ (respectively, $\alpha = -1$). Accordingly, the α-*map* has the opposite effect on the bit rate of inter-frame coding, even though the change is relatively modest.

Fig. 5 further visualizes how the α-*map* impacts the motion bit rate and the quality of the compressed optical flow map patch-wisely. In this experiment, the α-*map* is divided into two halves. The left halve has a fixed α value of 0, while that of the right halve changes from 1 to -1 (from top to bottom). We see that the motion bit rate increases with the decreasing α value while the compressed optical flow map has increasing details. These results validate that our model reacts to the given α-*map* in the way that we want it to.

C. Rate-Distortion Performance

Table II presents the BD-rate comparison relative to the anchor, which sets the α-*map* uniformly to 0. The two variants (Ours[1] vs. Ours[2]) of the proposed method refer to optimizing the α-*map* by considering only the current frame and by additionally looking ahead to one future frame, respectively (See Section III-D). From Table II, both variants exhibit 2%-5% rate savings than the anchor, and generally outperform DCVC [8]. The look-ahead variant (Ours[2]) achieves slightly higher gain at the cost of higher buffering requirements. It is worth noting that the gain of these variants is most obvious on test sequences with fast motion, e.g. Jocky, ReadySteadyGo and RaceHorses. In comparison with the other sequences, they have higher motion bit rates. As such, adapting the motion bit rate to the content of these sequences exerts a more significant influence on the rate-distortion performance. The same observation also holds true for those challenging sequences selected from CLIC'22 [25] (CLIC-MIX*). Last but not least, it is seen that conducting content-specific coding optimization through the adaptation of the α-*map* consistently shows gain over different test sequences. This suggests that there exists a domain gap between training and individual sequences, and that our scheme is able to help reduce the gap.

Fig. 6 visualizes how the compressed optical flow map changes with the α-*map* optimization. In this example, the α-*map* has a tendency to be decreased in exchange for a higher motion rate to represent the complex structure inherent in the optical flow map. It seen that the resulting flow map preserves more details than the initial flow map compressed with $\alpha = 0$.

V. Conclusion

This paper presents a content-adaptive motion rate adaptation scheme for learned video compression. It uses a bit allocation map to trade off between the bit rates for motion

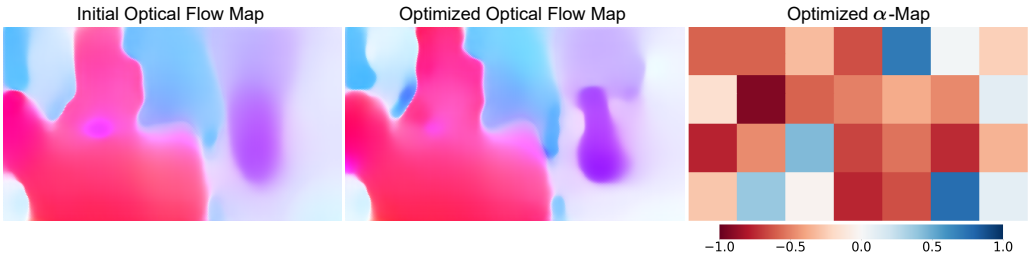


Fig. 6: Visualization of the initial optical flow map with $\alpha = 0$ (left) and the resulting optical flow map (middle) with the optimized α-*map* (right).

TABLE II: BD-rate comparison.

Datasets	ID	BD-rate (%) PSNR		
		DCVC	Ours[1]	Ours[2]
UVG*	Beauty	-0.88	-2.49	-2.77
	Bosphorus	-1.01	-2.12	-2.60
	HoneyBee	9.42	-1.36	-1.51
	Jockey	-3.19	-4.30	-4.61
	ReadySteadyGo	-8.91	-3.89	-4.07
	ShakeNDry	-15.83	-0.68	-1.40
	YachtRide	4.58	-3.12	-3.28
HEVC-D	BasketballPass	-0.70	-2.16	-3.17
	BlowingBubbles	5.49	-3.27	-3.19
	BQSquare	38.91	-2.47	-2.66
	RaceHorses	5.92	-3.39	-3.32
CLIC-MIX*	5d8f0 (zoom-in)	44.32	-2.04	-2.28
	25f0c (fast motion)	7.78	-1.59	-2.11
	4761c (rotation)	26.41	-4.49	-5.16
	a89f6 (zoom-out)	-4.09	-3.89	-3.98
	cda52 (shaking)	4.46	-2.46	-2.78

and inter-frame coding in a spatially-adaptive manner. The map is optimized through online back-propagation. Our major findings include: (1) content-adaptive motion rate adaptation helps mitigate the domain gap between training and test data; (2) the proposed scheme is found to be more effective on test sequences with complicated motion; and (3) our look-ahead mechanism is able to better optimize the bit allocation map for higher coding gain.

REFERENCES

[1] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "Dvc: An end-to-end deep video compression framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 006–11 015.

[2] G. Lu, X. Zhang, W. Ouyang, L. Chen, Z. Gao, and D. Xu, "An end-to-end learning framework for video compression," *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2020.

[3] J. Lin, D. Liu, H. Li, and F. Wu, "M-lvc: multiple frames prediction for learned video compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3546–3554.

[4] O. Rippel, A. G. Anderson, K. Tatwawadi, S. Nair, C. Lytle, and L. Bourdev, "Elf-vc: Efficient learned flexible-rate video coding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 14 479–14 488.

[5] E. Agustsson, D. Minnen, N. Johnston, J. Balle, S. J. Hwang, and G. Toderici, "Scale-space flow for end-to-end optimized video compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8503–8512.

[6] Z. Hu, G. Lu, and D. Xu, "Fvc: A new framework towards deep video compression in feature space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1502–1511.

[7] T. Ladune, P. Philippe, W. Hamidouche, L. Zhang, and O. Déforges, "Optical flow and mode selection for learning-based video coding," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2020, pp. 1–6.

[8] J. Li, B. Li, and Y. Lu, "Deep contextual video compression," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[9] Y.-H. Ho, C.-P. Chang, P.-Y. Chen, A. Gnutti, and W.-H. Peng, "Canf-vc: Conditional augmented normalizing flows for video compression," in *European Conference on Computer Vision*, 2022.

[10] J. Campos, S. Meierhans, A. Djelouah, and C. Schroers, "Content adaptive optimization for neural image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

[11] N. Zou, H. Zhang, F. Cricri, H. R. Tavakoli, J. Lainema, M. Hannuksela, E. Aksu, and E. Rahtu, "L 2 c–learning to learn to compress," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2020, pp. 1–6.

[12] G. Lu, C. Cai, X. Zhang, L. Chen, W. Ouyang, D. Xu, and Z. Gao, "Content adaptive and error propagation aware deep video compression," in *European Conference on Computer Vision*. Springer, 2020, pp. 456–472.

[13] T. van Rozendaal, J. Brehmer, Y. Zhang, R. Pourreza, and T. S. Cohen, "Instance-adaptive video compression: Improving neural codecs by training on the test set," *arXiv preprint arXiv:2111.10302*, 2021.

[14] H. Zhang, F. Cricri, H. R. Tavakoli, M. Santamaria, Y.-H. Lam, and M. M. Hannuksela, "Learn to overfit better: finding the important parameters for learned image compression," in *2021 International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2021, pp. 1–5.

[15] F. Brand, K. Fischer, and A. Kaup, "Rate-distortion optimized learning-based image compression using an adaptive hierachical autoencoder with conditional hyperprior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021, pp. 1885–1889.

[16] M. Song, J. Choi, and B. Han, "Variable-rate deep image compression through spatially-adaptive feature transform," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2380–2389.

[17] Z. Hu, G. Lu, J. Guo, S. Liu, W. Jiang, and D. Xu, "Coarse-to-fine deep video coding with hyperprior-guided mode prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5921–5930.

[18] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8934–8943.

[19] X. Wang, K. Yu, C. Dong, and C. C. Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 606–615.

[20] Y.-H. Ho, C.-C. Chan, W.-H. Peng, H.-M. Hang, and M. Domański, "Anfic: Image compression using augmented normalizing flows," *IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 613–626, 2021.

[21] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[22] J. B. Diederik P. Kingma, "Adam: A method for stochastic optimization," *International Conference for Learning Representations*, 2015.

[23] A. Mercat, M. Viitanen, and J. Vanne, "Uvg dataset: 50/120fps 4k sequences for video codec analysis and development," in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020, pp. 297–302.

[24] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[25] ""5th challenge on learned image compression"," *URL http://compression.cc*, 2022.