Learned Hierarchical B-frame Coding with Adaptive Feature Modulation for YUV 4:2:0 Content

Mu-Jung Chen[1] Hong-Sheng Xie[1] Cheng Chien[1] Wen-Hsiao Peng[1] Hsueh-Ming Hang[2]
[1]Computer Science Dept., [2]Electronics Engineering Dept., National Yang Ming Chiao Tung University, Taiwan

***Abstract*—This paper introduces a learned hierarchical B-frame coding scheme in response to the Grand Challenge on Neural Network-based Video Coding at ISCAS 2023. We address specifically three issues, including (1) B-frame coding, (2) YUV 4:2:0 coding, and (3) content-adaptive variable-rate coding with only one single model. Most learned video codecs operate internally in the RGB domain for P-frame coding. B-frame coding for YUV 4:2:0 content is largely under-explored. In addition, while there have been prior works on variable-rate coding with conditional convolution, most of them fail to consider the content information. We build our scheme on conditional augmented normalized flows (CANF). It features conditional motion and inter-frame codecs for efficient B-frame coding. To cope with YUV 4:2:0 content, two conditional inter-frame codecs are used to process the Y and UV components separately, with the coding of the UV components conditioned additionally on the Y component. Moreover, we introduce adaptive feature modulation in every convolutional layer, taking into account both the content information and the coding levels of B-frames to achieve content-adaptive variable-rate coding. Experimental results show that our model outperforms x265 and the winner of last year's challenge on commonly used datasets in terms of PSNR-YUV.**

***Index Terms*—video compression, YUV 4:2:0 format, variable rate, adaptive coding**

I. Introduction

Despite the fact that the recent developments of end-to-end learned video compression have shown promising coding performance, there remain many issues to be addressed. Most of the existing works [1]–[10] focus on P-frame coding, while B-frame coding, which allows the use of both the future and past reference frames for higher coding efficiency, is largely under-explored. Among the others, B-frame coding needs to address excessive motion overhead and the efficient use of the two reference frames. Besides, most learned codecs operate internally in the RGB 4:4:4 domain, even when the input is YUV 4:2:0 content. The conversation from YUV 4:2:0 to RGB 4:4:4 is introduced before compression to tackle the uneven spatial resolutions of the Y (luminance) and UV (chrominance) components. It is also common that multiple networks are used to achieve variable-rate compression, which is impractical in many real-world applications.

There have been few attempts at B-frame coding. Wu *et al.* [11] perform contextual coding of a B-frame based on the motion-compensated, multi-scale features extracted from the two reference frames. The idea is coined conditional (inter-frame) coding in [12], which additionally introduces one-stage, conditional motion coding without estimating flow maps explicitly. Following a more traditional approach, the works in [6], [13], [14] first encode two optical flow maps derived from the past and future references, followed by synthesizing a predicted frame for residual coding in the feature or pixel domain. To reduce motion overhead, Pourreza *et al.* [15] interpolate a predicted frame as a reference frame for coding a B-frame with the P-frame codec.

To achieve YUV 4:2:0 coding, Egilmez *et al.* [16] propose a branched network that processes Y and UV components separately before fusing their latents into a combined representation for coding. The other straightforward approaches include applying the space-to-depth conversion of the Y component, or coding Y and UV components separately. The latter calls for separate Y and UV codecs. These ideas are studied for image compression only.

To achieve variable-rate compression, it is common to introduce conditional convolution to adapt feature distributions [17]–[19] according to a rate-dependent hyperparameter. Lately, the idea of conditional convolution is extended to video coding [20] and [21], to achieve variable-rate compression with a single model.

In response to the Grand Challenge on Neural Network-based Video Coding at ISCAS 2023 [22], we propose a hierarchical B-frame coding system for YUV 4:2:0 content. Inspired by [23], we adopt conditional augmented normalizing flows (CANF) to perform conditional motion and inter-frame coding. In particular, we encode the Y and UV components by two separate conditional codecs, where the coding of the UV components is conditioned additionally on the Y component. Moreover, to achieve content-adaptive variable-rate coding with a single model, we extend conditional convolution to accommodate not only the rate parameter, but also the coding levels of B-frames and the output features of the convolutional layers. Experimental results show that our model outperforms x265 [24] and the winner [21] of last year's challenge [25] on commonly used datasets in terms of PSNR-YUV.

II. Related Work

This section reviews the basics of the CANF-based inter-frame coding [23], to ease the understanding of our scheme. Fig. 1 illustrates its architecture for inter-frame coding, which aims to learn the conditional distribution $p(x_t|x_c)$ of the coding frame x_t given its motion-compensated reference frame x_c. In [23], this is achieved by maximizing the conditional augmented likelihood $p(x_t, e_z, e_h|x_c)$, where e_z, e_h are the two augmented noise inputs. From the operational perspective, x_c serves as the conditioning factor in the autoencoding transforms, composed of $\{g_{\pi_i}^{enc}, g_{\pi_i}^{dec}\}_{i=1}^{2}$, which convert (from left to right) x_t into x_c, with the quantized latent $\hat{z}_2$ capturing the information needed to instruct the conversion and $\hat{h}_2$ taking

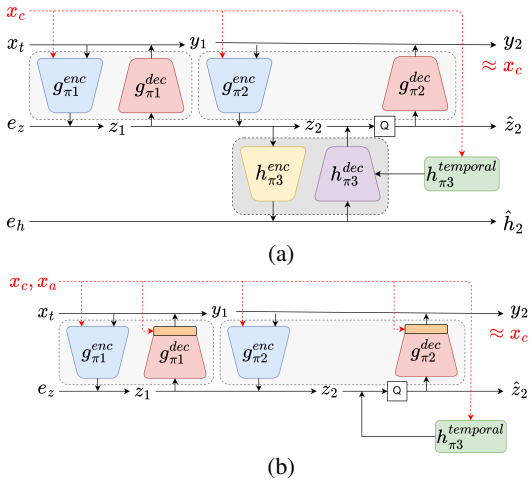


Fig. 1: Illustration of (a) the CANF-based inter-frame codec, which is used for coding the Y component in this work, and (b) the CANF-based inter-frame codec for coding the UV components, where x_a represents the coded Y component.

the role of the hyperprior. The conversion between x_t and x_c is approximate and lossy. The same CANF-based codec can also be utilized to encode optical flow maps conditionally. To this end, a flow map predictor must be created to serve as the condition.

III. Proposed Method

A. System Overview

Fig. 2 presents an overview of our proposed method. As shown, the encoding of a B-frame x_t^{420} begins with using the motion estimation network (MENet) operating internally in the YUV 4:4:4 domain to obtain bi-directional optical flow maps $m_{t\to t-k}, m_{t\to t+k}$ according to its two reference frames $\hat{x}_{t-k}^{420}, \hat{x}_{t+k}^{420}$, respectively. The resulting flow maps are compressed jointly by the CANF-based conditional motion codec (M, M^{-1}) given the conditioning signals $m_{t\to t-k}^p, m_{t\to t+k}^p$ generated by the motion prediction network (MPNet). The decoded flow maps $\hat{m}_{t\to t-k}, \hat{m}_{t\to t+k}$ are used for bi-directional motion compensation. Particularly, we adopt two separate motion compensation networks (MCNet-*Y*, MCNet-*UV*) to synthesize the motion-compensated frames $\hat{x}_c^y, \hat{x}_c^{uv}$ for Y and UV components, respectively. $\hat{x}_c^y, \hat{x}_c^{uv}$ serve as the conditioning signals for conditional inter-frame coding of x_t^y, x_t^{uv} to obtain the reconstructed Y and UV components $\hat{x}_t^y, \hat{x}_t^{uv}$, respectively. Notably, for coding the UV components, we introduce the reconstructed Y component as an additional conditioning signal. The following sections elaborate on these proposed modules.

B. Conditional Bi-directional Motion Coding

The proposed conditional motion codec follows the same CANF-based design as Fig. 1 (a). In the present context, we concatenate $m_{t\to t-k}, m_{t\to t+k}$ to form a 4-channel input for coding (i.e. x_t in Fig. 1 (a) becomes the concatenated signal from $m_{t\to t-k}, m_{t\to t+k}$). Likewise, the predicted flow maps $m_{t\to t-k}^p, m_{t\to t+k}^p$ output by the motion prediction network

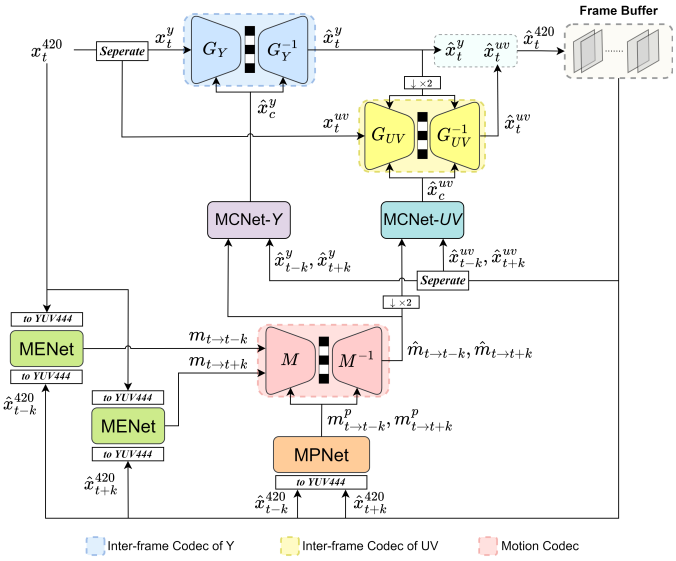


Fig. 2: The proposed B-frame coding framework. x_t^{420} denotes the current coding frame and $\hat{x}_{t-k}^{420}, \hat{x}_{t+k}^{420}$ are the two previously reconstructed reference frames. "*Separate*" is an operation that separates the Y and UV components. "$\downarrow \times 2$" is the downsampling operation by a factor of 2. Note that in downsampling an optical flow map (m), the values of the flow map are also reduced by half.

(MPNet) are concatenated to replace x_c in Fig. 1 (a) as the conditioning signal for motion coding. This allows the motion codec to exploit freely the correlation inherent in $m_{t\to t-k}, m_{t\to t+k}, m_{t\to t-k}^p$, and $m_{t\to t+k}^p$.

C. Motion Estimation and Compensation Networks

As illustrated in Fig. 2, motion estimation and motion compensation are done by MENet and MCNet, respectively. In an effort to re-use PWCNet [26] without introducing any significant change, we interpolate the UV components and fine tune the PWCNet [26] to perform motion estimation in the YUV 4:4:4 domain. The flow maps thus obtained have the same resolution as the Y component and are downscaled to motion compensate the UV components.

For motion compensation, separate MCNets are used to process Y and UV components distinctively. Similar to the motion compensation network in [1], our MCNet includes as inputs the Y/UV components from the future and past reference frames, and the decoded flow maps, which are used for bi-directional backward warping. Notably, we reduce the number of channels from 64 to 48 to avoid an excessive increase in model size due to the use of two MCNets.

D. Conditional Inter-frame Coding in the YUV Domain

Two conditional inter-frame codecs are used to code the Y and UV components separately. They follow the CANF-based coding schemes in Figs. 1 (a) and (b), respectively, where x_c denotes the motion-compensated Y or UV components. Because the Y component preserves most of the information, we use the decoded Y component $\hat{x}_t^y$ as an additional conditioning

signal x_a in Fig. 1 (b) when coding the UV components. Notably, we remove the hyperprior [27] from the UV codec and use only the temporal prior [3], [9] to reduce complexity. A side experiment shows that this design choice has little impact (<1% rate inflation) on coding performance.

E. Adaptive Feature Modulation

Adaptive feature (AF) modulation is to adapt the feature distribution in every convolutional layer, in order to achieve variable-rate compression with a single model and content-adaptive coding. The AF modulation is placed after every convolutional layer in the motion and inter-frame codecs. As shown in Fig. 3, it outputs channel-wise affine parameters, which are used to dynamically adjust the output feature distributions.

As compared to the previous works [20], [21], our scheme has two distinctive features. One is that we introduce the coding level C of a B-frame as its contextual information to achieve hierarchical rate control. This is motivated by the fact that with hierarchical B-frame coding, the reference quality of a B-frame varies with its coding level. The additional contextual information from the coding level allows greater flexibility in adjusting the bit allocation among B-frames. We note that most previous works use only a single rate parameter λ as the contextual information without distinguishing between B-frames of different coding levels. Additionally, our AF module incorporates a global average pooling (GAP) layer to summarize the input feature maps with a 1-D feature vector. As such, our AF module is able to adapt the feature distribution in a content-adaptive manner.

In our current implementation, C=0,1 has only two values because during training, there are only a limited number of coding levels (see Sec. III-F). C=0 means the current coding B-frame will serve as a reference frame for the other B-frames at higher coding levels, while C=1 indicates that it is at the highest coding level and will not be utilized for reference. See Table I for an example. Moreover, we choose $\lambda \in \{16384, 4096, 1024, 256, 128\}$ to encode B-frames at a finite number of rate points. To achieve fine-grained rate control, we incorporate an intra codec that supports continuous-step rate adaptation. To encode a video sequence at a specific rate point, we first choose from a set of pre-determined combinations of λ's for the intra and inter codecs the one which yields a rate point matching closely the target rate. We then fine tune the λ of the intra codec to fit the target rate precisely.

F. Training Loss

Our training objective function is given by

$$L = \frac{1}{5}\sum_t \lambda \cdot [6d(x_t^y, \hat{x}_t^y) + 2d(x_t^{uv}, \hat{x}_t^{uv})]/8 + R_t, \quad (1)$$

where t is the frame index, $d(\cdot,\cdot)$ measures the mean-squared error between the input and the output Y/UV components (the weighting factors 6 and 2 follow the evaluation metrics of the challenge [22]), and R_t is the bit rate consumed by all the codecs. We train our scheme on 5-frame training sequences,

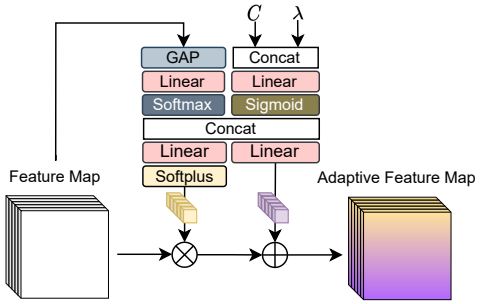


Fig. 3: Adaptive feature (AF) modulation for content-adaptive variable-rate coding. C, λ denote the coding level and the rate parameter, respectively. $\otimes, \oplus$ are the channel-wise multiplication and addition, respectively.

TABLE I: 5-frame hierarchical B-prediction

t	1	2	3	4	5
frame-type	I	B2 (C=1)	B1 (C=0)	B2 (C=1)	I

each of which is encoded as two-level hierarchical B-frames with an intra-period 4 (see Table I).

IV. Experiments

A. Setup

We train our model on Vimeo-90k [28], where the video sequences are of size 448x256 in RGB format. For training, we randomly crop video frames to 256x256 and convert their color space from RGB to YUV 4:2:0. We use Adam optimizer [29], with the learning rate set to $1e^{-4}$.

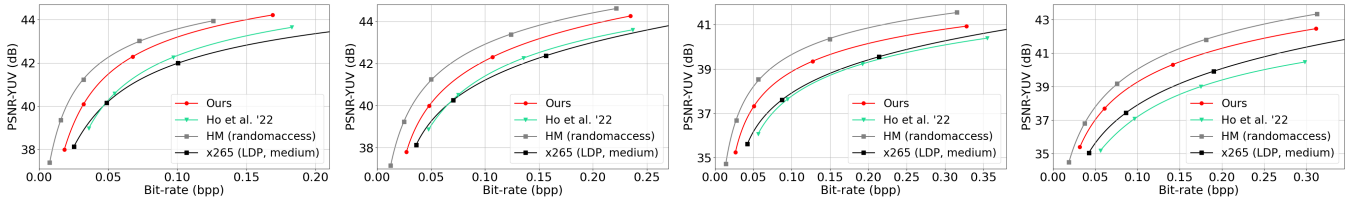
For intra coding, we adopt a similar intra codec to [30]. We, however, remove the context model and replace the Gaussian Mixture model with a simple Gaussian to reduce the coding runtime.

The evaluation of compression performance is done on UVG [31], MCL-JVC [32], HEVC Class B [33] and ISCAS'22 Grand Challenge (GC) [25]. All the test sequences are in YUV 4:2:0 format. The intra-period defaults to 32. We follow [25] to calculate the PSNR, with the bit rate measured in bits per pixel (bpp).

The proposed method is compared against x265 [24] under *medium* preset and *low delay* configuration, and HM [34] with *encoder_randomaccess_main* configuration. The former is used as anchor in reporting the BD-rate numbers, unless otherwise specified. Additionally, the learned baseline method is [21], which is the top performer in ISCAS 2022 challenge [25]. Note that [21] supports P-frame coding only. We remark that none of the existing learned B-frame codecs supports YUV 4:2:0 content.

B. Experimental Results

Fig. 4 shows the rate-distortion comparison, with the BD-rate numbers reported in Table II. Two observations are immediate. First, our method outperforms x265 and the learned codec in [21] by a large margin across all the datasets. This is attributed to the use of more efficient B-frame coding. Second, the proposed method is seen to be inferior to HM



(a) UVG (b) MCL-JCV (c) HEVC Class B (d) ISCAS'22 GC

Fig. 4: Rate-distortion plots on UVG, MCL-JCV, HEVC Class B, and ISCAS'22 GC datasets in terms of PSNR-YUV.

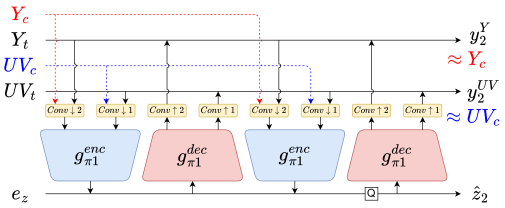


Fig. 5: Illustration of the merged YUV coding. For brevity, the hyperprior and the temporal prior are omitted from the figure.

under the random access configuration, which represents a much stronger baseline method for B-frame coding. In contrast to our comparison, the recent work [14] compares with HM randomaccess in terms of PSNR-RGB with a short intra-period of 8. We note that their experimental settings work favorably to learned codecs. To evaluate the decoded picture quality in the RGB domain, the pipeline of learned compression typically involves the color space conversion from YUV 4:2:0 as the input format to RGB for encoding and decoding. This is to be compared with the setting of HM, namely, encoding and decoding in YUV 4:2:0, followed by the color space conversion to RGB as the output format. As such, learned codecs are able to leverage end-to-end training to maximize the decoded picture quality in RGB. To compare fairly with HM, the ISCAS 2023 challenge requires that the quality evaluation be done in YUV 4:2:0.

Table III presents the BD-rate comparison between our conditional YUV coding scheme and its variants, including (1) *independent* coding of the Y and UV components with two separate conditional inter-frame codecs, (2) *merged* coding of the Y and UV components by converting them into their latent representations using convolutional layers for concatenation and joint coding with one conditional inter-frame codec (Fig. 5), (3) *space-to-depth* conversion of the Y component for concatenation with the UV components and their joint coding with one conditional inter-frame codec, and (4) *YUV 4:4:4* conversion from YUV 4:2:0 for joint coding of the Y and UV components with one conditional inter-frame codec. For fair comparison, these competing methods have a similar model size. The BD-rate numbers are reported for the Y, U, and V components separately. We observe that our method outperforms all the other variants across different color components, except for the independent case, where the Y component shows slightly more rate saving (2.3%) at the cost of much worse compression performance on the U and V components. The gain of the Y component is related to the bit allocation among these color components. Generally, the independent case suffers from much worse compression performance on the UV components.

TABLE II: BD-rate comparison. The anchor is x265 in LDP medium mode.

Method	BD-rate (%) PSNR			
	UVG	MCL-JCV	HEVC-B	ISCAS'22 GC
Ours	-36.6	-26.6	-32.5	-33.4
Ho *et al.*'22 [21]	-10.9	-4.5	7.7	24.9
HM [34]	-57.0	-48.7	-53.8	-48.2

TABLE III: Ablation study on YUV coding. The dataset is ISCAS'22 GC. The anchor is our proposed method.

YUV Coding	BD-rate (%) PSNR		
	Y	U	V
Ours	0.0	0.0	0.0
Independent	-2.3	4.0	17.5
Merged	15.1	81.3	82.3
Space-to-Depth	10.3	96.2	51.0
YUV444	4.8	57.4	53.9

TABLE IV: Ablation study on adaptive feature modulation. The anchor is our method with fully functional modulation.

Content Adaptive	Coding Level	BD-rate (%) PSNR			
		UVG	MCL-JCV	HEVC-B	ISCAS'22 GC
✓	✓	0.0	0.0	0.0	0.0
✓		5.4	5.0	5.1	8.2
	✓	12.3	9.6	10.0	10.2
		16.0	13.3	13.9	17.4

Table IV presents an ablation study of our adaptive feature modulation, with the aim of understanding the gain from the two pieces of contextual information, i.e. the content feature and the coding level C of the target frame (see Fig. 3). It is seen that disabling either of them or both leads to considerable rate inflation.

Conclusion

We propose a hierarchical B-frame coding scheme with adaptive feature module for YUV 4:2:0 content. Our major findings include: (1) separately coding the Y and UV components is beneficial, (2) adapting the feature distributions to the content information and the coding levels of B-frames is crucial to content-adaptive variable-rate coding, and (3) in terms of coding YUV 4:2:0 content, our learned codec still has ample room for further improvement as compared to HM.

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] 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.

[3] H. Liu, M. Lu, Z. Ma, F. Wang, Z. Xie, X. Cao, and Y. Wang, "Neural video coding using multiscale motion compensation and spatiotemporal context model," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[4] 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.

[5] 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.

[6] R. Yang, F. Mentzer, L. V. Gool, and R. Timofte, "Learning for video compression with hierarchical quality and recurrent enhancement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6628–6637.

[7] Z. Hu, Z. Chen, D. Xu, G. Lu, W. Ouyang, and S. Gu, "Improving deep video compression by resolution-adaptive flow coding," in *European Conference on Computer Vision*. Springer, 2020, pp. 193–209.

[8] 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.

[9] R. Yang, F. Mentzer, L. Van Gool, and R. Timofte, "Learning for video compression with recurrent auto-encoder and recurrent probability model," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 388–401, 2020.

[10] A. Golinski, R. Pourreza, Y. Yang, G. Sautiere, and T. S. Cohen, "Feedback recurrent autoencoder for video compression," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[11] C.-Y. Wu, N. Singhal, and P. Krahenbuhl, "Video compression through image interpolation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[12] T. Ladune, P. Philippe, W. Hamidouche, L. Zhang, and O. Déforges, "Conditional coding for flexible learned video compression," in *Neural Compression: From Information Theory to Applications–Workshop@ ICLR 2021*, 2021.

[13] A. Djelouah, J. Campos, S. Schaub-Meyer, and C. Schroers, "Neural inter-frame compression for video coding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[14] M. A. Yılmaz and A. M. Tekalp, "End-to-end rate-distortion optimized learned hierarchical bi-directional video compression," *IEEE Transactions on Image Processing*, vol. 31, pp. 974–983, 2022.

[15] R. Pourreza and T. Cohen, "Extending neural p-frame codecs for b-frame coding," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6660–6669, 2021.

[16] H. E. Egilmez, A. K. Singh, M. Coban, M. Karczewicz, Y. Zhu, Y. Yang, A. Said, and T. S. Cohen, "Transform network architectures for deep learning based end-to-end image/video coding in subsampled color spaces," *IEEE Open Journal of Signal Processing*, vol. 2, pp. 441–452, 2021.

[17] J. L. Yoojin Choi, Mostafa El-Khamy, "Variable rate deep image compression with a conditional autoencoder," in *International Comference on Computer Visions*, 2019.

[18] Z. Cui, J. Wang, B. Bai, T. Guo, and Y. Feng, "G-vae: A continuously variable rate deep image compression framework," 03 2020.

[19] M. Song, J. Choi, and B. Han, "Variable-rate deep image compression through spatially-adaptive feature transform," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, oct 2021, pp. 2360–2369.

[20] J. L. H. L. F. W. Jianping Lin, Dong Liu, "A deeply modulated scheme for variable-rate video compression," in *IEEE International Conference on Image Processing*, 2021.

[21] Y.-H. Ho, C.-H. Lin, P.-Y. Chen, M.-J. Chen, C.-P. Chang, W.-H. Peng, and H.-M. Hang, "Learned video compression for yuv 4:2:0 content using flow-based conditional inter-frame coding," 2022.

[22] "The second grand challenge on neural network-based video coding with iscas 2023," *URL https://iscas2023.org/authors/call-for-grand-challenge*, 2022.

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

[24] "Ffmpeg," https://www.ffmpeg.org/, accessed: 2022-05-018.

[25] "Grand challenge on neural network-based video coding," *URL https://www.iscas2022.org/grand-challenge*, 2021.

[26] 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.

[27] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," *Advances in Neural Information Processing Systems*, vol. 31, pp. 10 771–10 780, 2018.

[28] 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.

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

[30] Y.-H. Ho, C.-C. Chan, W.-H. Peng, H.-M. Hang, and M. Domanski, "Anfic: Image compression using augmented normalizing flows," *arXiv preprint arXiv:2107.08470*, 2021.

[31] 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.

[32] H. Wang, W. Gan, S. Hu, J. Y. Lin, L. Jin, L. Song, P. Wang, I. Katsavounidis, A. Aaron, and C.-C. J. Kuo, "Mcl-jcv: a jnd-based h. 264/avc video quality assessment dataset," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1509–1513.

[33] F. Bossen *et al.*, "Common test conditions and software reference configurations," *JCTVC-L1100*, vol. 12, no. 7, 2013.

[34] "Hm reference software for hevc," https://vcgit.hhi.fraunhofer.de/jvet/HM/-/tree/HM-16.23/, accessed: 2022-05-018.