IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 May 2023, accepted 27 May 2023, date of publication 6 June 2023, date of current version 14 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3283277

RESEARCH ARTICLE

Hierarchical Random Access Coding for Deep Neural Video Compression

NGUYEN VAN THANG[1,2] **AND LE VAN BANG**[1]
[1]AI Camera Center, Viettel High Technology Industries Corporation, Hanoi 100000, Vietnam
[2]College of Engineering and Computer Science, VinUniversity, Hanoi 10000, Vietnam

Corresponding author: Nguyen Van Thang (thang.nv@vinuni.edu.vn)

This work was supported by Viettel High Technology Industries Corporation.

ABSTRACT Recently, neural video compression networks have obtained impressive results. However, previous neural video compression models mostly focus on low-delay configuration with the order of display being the same as the order of coding. In this paper, we propose a hierarchical random access coding approach that exploits bidirectionally temporal redundancy to improve the coding efficiency of existing deep neural video compression models. The proposed framework applies a video frame interpolation network to improve inter-frame prediction. In addition, a hierarchical coding structure is also proposed in this paper. Experimental results show the proposed framework improves the coding efficiency of the base deep neural model by 48.01% with the UVG dataset, 50.96% with the HEVC-class B dataset, and outperforms the previous deep neural video compression networks.

INDEX TERMS Neural video compression, hierarchical random access coding, video frame interpolation.

I. INTRODUCTION

Video coding, also known as video compression, is a crucial research topic, that enables efficient storage and transmission of video content. With a long history of research and standardization, various video coding standards such as AVC/H.264 [21], and HEVC/H.265 [1] are built and deployed successfully. Those video coding standards follow a hybrid approach that uses various advanced coding techniques such as variable block sizes, intra-frame prediction, inter-frame prediction, transform coding, quantization, and entropy coding to improve compression efficiency. This block-based hybrid approach is still considered the main component for the development of the next video standards.

Recently deep neural video compression obtains impressive results in terms of coding efficiency [10], [11], [12], [13]. Residual coding-based deep neural video compression methods apply inter-prediction coding from traditional hybrid video codecs [1], [2], [21], [22]. In detail, they use motion estimation and motion-compensated prediction to generate the residual between the predicted frame and the current encoding frame. Then, the residual, and the estimated motion vector field are encoded and transmitted to the decoder via entropy coding. The classical hybrid video coding such as AVC/H.264 [21], and HEVC/H.265 [1] are typically encoded with one of three coding configurations: all-intra configuration, low-delay configuration, and random access configuration. For deep neural video compression, two former coding configurations are researched intensively with outstanding performance in terms of coding efficiency [11], [12]. In all-intra coding configuration, each frame is encoded using a neural image compression model separately, it does not exploit the temporal redundancy between frames that are the most powerful coding tool in classical video coding with the highest contribution to coding efficiency of the classical video codecs, thanks to temporal inter-frame prediction. For low delay configuration, several state-of-the-art neural video compression models [11], [12], [13], [14], [15], [16] exploit inter-frame prediction with motion estimation and motion compensation networks. However, they just apply inter-frame prediction at one side of the current coding frame, the previous frame, or P-frame prediction, due to the low delay requirement of encoding processing. As proved by classical video coding standards [1], [2], [21], [22], random access configuration is the most efficient one to improve coding efficiency compared to all-intra and low-delay coding

The associate editor coordinating the review of this manuscript and approving it for publication was Chaker Larabi.

configurations, thanks to its B-frame prediction that exploits temporal redundancy at both directions of the current encoding frame, the previous frame and the future frame. In this paper, we propose a novel deep neural video compression model for random access configuration that is easy to extend from previous low-delay deep neural video compression models that are researched extensively in the field. In addition, in random access configuration, a hierarchical coding scheme is applied to allocate effectively bitrate for frames. In previous neural video compression models, they treat and allocate bitrate for each frame equally. Consequently, it causes a high bit rate due to uniform bitrate allocation. In random access configuration, we can allocate low bit-rate for frames at high levels of Group of Picture (GoP) because they are not used as a reference frame for encoding other frames. Therefore the quality of those reconstructed frames does not affect the encoding processing of other frames. In this paper, we borrow the hierarchical coding structure from the classical video coding standard into the deep neural video compression model. In addition, in this paper, we also propose to use a video frame interpolation model in the inter-prediction mode of hierarchical random access configuration to improve the coding efficiency of the network. In summary, the primary contributions of the proposed method are as follows:

- A random access coding for a deep neural video compression that exploits temporal redundancy and improves coding efficiency significantly in comparison with other configurations.
- A hierarchical coding scheme that allocates bitrate for each frame depending on its level at a group of pictures.
- A video frame interpolation model is applied to improve inter-frame prediction.
- Experimental results demonstrate that the proposed method outperforms the previous neural video compression models significantly in terms of coding efficiency. In addition, this model is easy to extend to previous P-frame neural video compression models.

II. RELATED WORK

A. NEURAL VIDEO COMPRESSION

Deep neural networks (DNNs) based video compression has attracted extensive attention from improving existing coding tools [3], [5], [8], [9], [31] to end-to-end neural video coding [4], [6], [7], [10], [11], [12], [13], [14], [15], [16]. In [9], Khani et al. proposed a video coding pipeline via content-adaptive super-resolution where both sequence bitstream and model bitstream are transmitted from encoder to decoder. In [31], Gang et al. proposed a hybrid video coding system that consists of an adaptive overfitted multi-scale attention network and a classical video codec. In [10] Mentzer et al. proposed a transformer-based video compression model. They used a transformer to model temporal dependencies between input frames. Consequently, the method retains the intrinsic drawbacks of the transformer model which are high complexity and long-range dependency. In [12] Li et al. proposed to use a hybrid spatial-temporal entropy modeling for video compression. The method inherited a context modeling proposed in [11]. In the other approach, Hu et al. [6] applied coarse-to-fine coding with hyperprior-guided mode prediction, and Agustsson et al. [16] proposed scale-space flow instead of an explicit motion estimation network for end-to-end video compression. Pourreza et al. [15] improves the coding efficiency of a neural video compression model by exploiting hierarchical redundancy between frames for both residual and flow information. Sun et al. [34] proposed a spatiotemporal entropy model for learned video compression. However, the previous above-mentioned methods mostly focus on low delay configuration where the coding order of frames is the same as the display order of the video sequence.

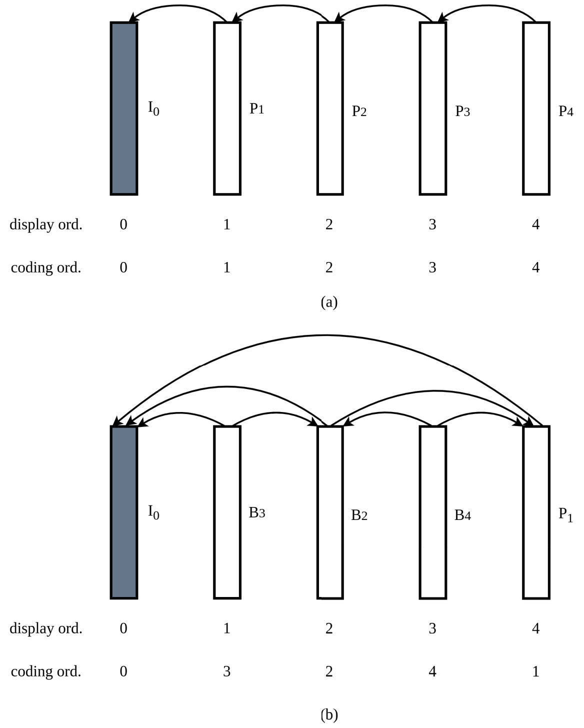
B. B-FRAME CODING WITH FRAME INTERPOLATION

Video frame interpolation obtains impressive results with deep networks from kernel-based separated adaptive convolution networks [26], [27], [28] to transformer-based networks [29], [30]. Several previous works exploit the recent advantage of the frame interpolation problem in neural video compression. Chao et al. [32] considered video compression as repeated image interpolation, therefore it compressed some frames first as image compression, and use them as key-frames to interpolate in-between frames via a frame interpolation network. Pourreza and Cohen [33] extended P-frame coding from low-delay to B-frame coding by exploiting a super-slo-mo video frame interpolation network [31]. Nguyen et al. [28] proposed a stacked video frame interpolation network to enhance the visual quality of compressed video encoded via HEVC. However, previous methods either focus on the Video Frame Interpolation (VFI) task or apply directly the VFI models trained for the video frame interpolation task into compressed videos, therefore the contributions of VFI models are limited. To avoid the limitations, in this paper, we apply a video frame interpolation network that is trained on compressed images to improve the coding efficiency of the neural video compression model.

III. PROPOSED METHOD

A. BASELINE MODELS FOR I-FRAME AND P-FRAME

I-frame and P-frame codecs are shown in Figure 2a, and 2b respectively. The I-frame coding network consists of a single autoencoder, denoted as Image-AE that compresses input frame f_t to a compressed output image $f*_t$, meanwhile the P-frame coding network generates a predicted frame, denoted as $f*_{pred}$ of f_t by using a scale-space flow estimation, marked as Motion AE, and motion compensation via the scale-space flow-based Warping operations and then adding residuals into the predicted frame to generate the final reconstructed frame that will become the reference picture to encode for next frame. In this paper, we apply the SSF model [16] for the I-frame and P-frame coding.



(a)

(b)

FIGURE 1. (a): Low delay coding; (b): Hierarchical random access B coding structure in video coding.

B. VIDEO FRAME INTERPOLATION

Given two consecutive frames I_0, and I_1 of a video sequence, a video frame interpolation generates an intermediate frame I_t, where t ∈ (0, 1). Among various methods for frame interpolation, we select IFRNet [18] which achieves competitive accuracy with fast inference speed and lightweight model size. In addition, the IFRNet is able to produce multiple interpolated frames that can be helpful for video compression.

However, unlike existing methods that typically train a video frame interpolation network using the uncompressed original images of the Vimeo90K dataset [19], our paper employs compressed frames generated with the JPEG codec [20] to produce various artifacts in compressed images. Because JPEG and video codecs such as H.264/AVC, and H.265/HEVC use a block-based approach for coding, they exhibit similar characteristics and artifacts such as blocking, ringing, and ghost artifacts. Previous methods have utilized a video frame interpolation model that is pretrained and specific to the video frame interpolation task, which makes it unable to learn the distorted features of compressed images. To the best of our knowledge, our method is the first to use compressed images for video frame interpolation or frame rate up-conversion tasks. This strategy will be useful for future research exploring the application of frame rate up-conversion models to video compression.

C. RANDOM ACCESS CONFIGURATION IN NEURAL VIDEO CODING

In random access configuration, the order of encoding time is different from that of display time. That means future frames in display time order can be encoded before the current frame. Consequently, the future frames can become reference pictures during the encoding process of the current frame. In a Group of Pictures (GoP), the order of encoding is predefined. Figure 1 shows an example of a coding structure of a GOP with five frames in low delay (a) and random access configurations (b). Previous neural video compression models focus on improving the coding efficiency of low-delay configuration where the coding order is the same as the display order of the video sequence and the coding of the current frame depends only on the encoded previous frame. Vice versa, in this paper with a hierarchical-B coding structure, the coding order is different from the display order. Consequently, each encoding frame has three reference pictures, one is the past picture, and the other is the future picture, both pictures are reconstructed before encoding the current frame, and the last one is a combinative picture generated from the two former pictures via a frame interpolation network described in section B. Then, all three frames are used as the reference pictures of the current encoding frame. Figure 1b shows an example of the hierarchical-B coding structure of a GoP with five frames. The start point of the arrows represents the current encoding frame while the endpoint of the arrows represents the referenced frames during encoding. In addition, in the hierarchical-B coding structure, rate distortion tradeoff parameters (lambda) are changed hierarchically during encoding frames in GoP. Figure 1b shows a GoP example with three levels, frame I_0, and P_1 belong to the lowest level, level 0, frame B_2 belongs to level 1, frame B_3 and frame B_4 belong to level 2, the highest level. Consequently, frames encoded with the highest level will not become reference pictures for other frames due to their low quality. In this example, frame B_3 and B_4 are non-referenced B pictures.

D. B FRAME PREDICTION NETWORK

In classical hybrid video coding such as HEVC/H.265, B-frame prediction uses both the previous reference frame and the future reference frame for motion estimation and motion compensation. Therefore, it needs to transmit bidirectional motion vectors for each block. Consequently, it requires more bits to encode motion information. To reduce numbers of bits for encoding motion information, we propose to use a video frame interpolation (or frame rate up-conversion) network to generate a representative reference picture of both the previous reference frame and the future reference frame but only need to encode a unidirectional motion vector. This reduces the bitrate for motion coding. In addition, the generated intermediate frame interpolated from the two reference pictures is the closest to the current encoding frame. Therefore, the residual is the smallest, and mostly zero. It also improves the coding efficiency of residual coding. In the video frame interpolation, from two reconstructed references, denoted as f^*_0, and f^*_1, the network interpolates the intermediate frame at time t where t ∈ (0, 1). In fact, t could be anywhere between 0 and 1, but as mentioned above, in order to improve coding efficiency,

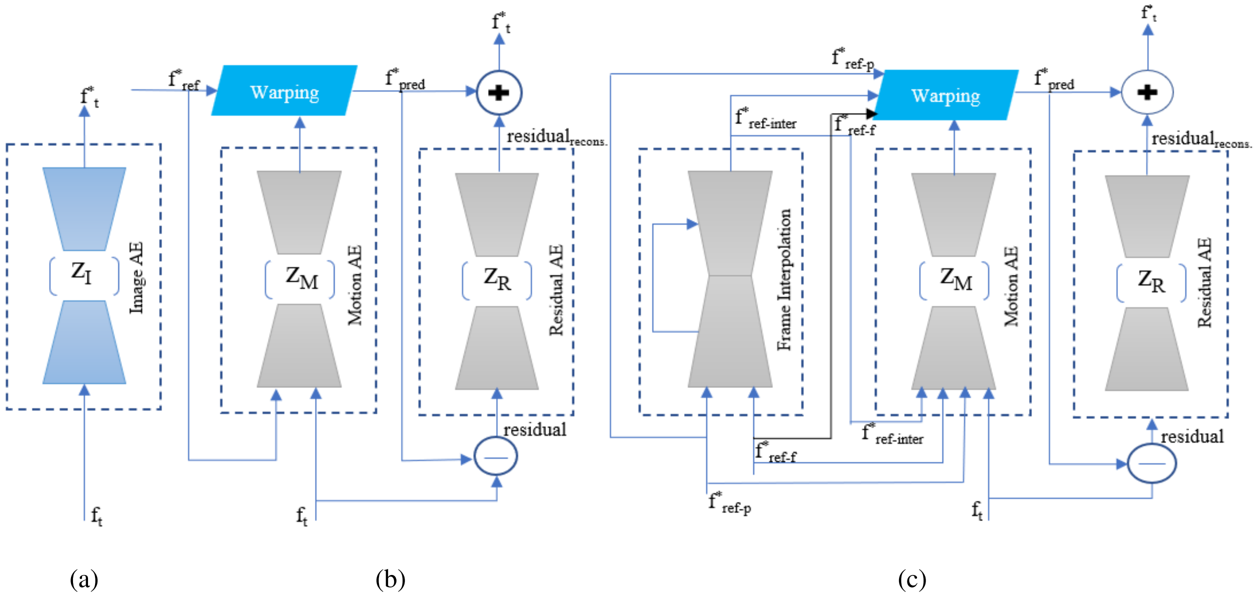


FIGURE 2. (a) I-frame coding network and (b) P- frame coding network based on SSF [2], (c) The proposed hierarchical random access coding network. All are components of our neural video compression model.

it often selects t = 0.5 for symmetrical references, though it will impose restrictions on the GoP size in random access configuration. We apply IFRNet [18] as a frame interpolator to generate a combinative reference frame from two previous and future reference pictures. Then we adapt Scale-Space Flow (SSF) [16] as a prediction model that generates the residuals from the differences between the current encoding frame and the reference frames, including the generated combinative reference frame. In our framework, except the beginning frame and the end frame of each GoP, all other frames in GoP are encoded by B-prediction with support from the video frame interpolation network. In the example GoP with the size of 5, the coding type is $I_0B_3B_2B_4P_1$. In the other words, frame B_2, B_3, and B_4 are coded with B frame prediction via the IFRNet, as shown by the first block, marked as Frame Interpolation in Figure 2c and following expressions:

$$B^*_{2-ref-inter} = IFRNet(I^*_0, P^*_1) \tag{1}$$

$$B^*_{3-ref-inter} = IFRNet(I^*_0, B^*_2) \tag{2}$$

$$B^*_{4-ref-inter} = IFRNet(B^*_2, P^*_1) \tag{3}$$

where $B*_{2\text{-}ref\text{-}inter}$, $B*_{3\text{-}ref\text{-}inter}$, and $B*_{4\text{-}ref\text{-}inter}$ are the interpolated reference pictures when encoding B_2, B_3, and B_4 respectively. In addition, our B-frame coding's motion autoencoder takes four frames including the current encoding frame, and three reference pictures, as the input for implicit motion estimation and representations in latent space, the architecture of the motion autoencoder for B-frame is slightly modified to adapt more input frames. Unlike the previous methods with B-frame coding, the whole frame is encoded as B-frame, this modification is aligned to the classical video coding methods, when coding with B-frame, pixels, and regions still can be encoded as unidirectional P-slice coding or bidirectional B-slice coding. In other words, the motion autoencoder network learns to estimate motion vectors implicitly and pixels are predicted from one corresponding reference frame among three reference frames during training for rate-distortion optimization. Several pixels (or regions) are predicted from the predecessor reference P-frame, some other pixels (or regions) are derived from the successor reference P-frame, and the other pixels (or regions) are estimated from the interpolated reference B-frame. Which pixels are predicted from which reference frames, depends on the contents of frames and rate-distortion optimization during training.

E. END-TO-END HIERARCHICAL DEEP VIDEO COMPRESSION NETWORK

The proposed network contains three network components as shown in Figure 2. The first one is an I-frame network that encodes and decodes the very first frame, or Intra frame. The second network is a P-frame codec used for encoding and decoding the ending frame of each periodic GoP. For I-frame and P-frame codecs, we adapt the Scale-space flow architecture from [16]. The last one is the proposed hierarchical B-frame network to encode the other frames in each GoP with random access coding. As shown in Figure 2c, in order to encode the current frame, denoted f_t, two reference frames are required, one is the reconstructed previous frame, denoted as $f*_{ref\text{-}p}$ and the other is the reconstructed future frame, denoted as $f*_{ref\text{-}f}$. The two reference frames are inputs for a frame interpolation network to generate an intermediate reference frame, denoted as $f*_{ref\text{-}inter}$. Then the generated reference frame, $f*_{ref\text{-}inter}$ together with the previous reference frame, $f*_{ref\text{-}p}$, the future reference frame $f*_{ref\text{-}f}$,

TABLE 1. Rate-distortion trade-off values of frames during the training in a hierarchical B-frame coding with GoP of 5.

Frame	I	P_1	B_2	B_3	B_4
Lambda	λ_0	λ_0	λ_1	λ_2	λ_2

and the current frame f^*_t are inputs for a motion autoencoder (denoted as Motion AE in Figure 2c). This is a novel approach in comparison with previous methods that only use one reference picture, usually the previous reference picture or the generated reference picture for inter-frame prediction. We use all three frames to exploit coding efficiency from all sides, unidirectional inter-prediction from the previous frame, unidirectional inter-prediction from the future frame and bidirectional inter-prediction from the generated intermediate frame. Then a predicted frame is produced by warping the reference frames via a decoded motion field. For warping, we use scale-space-based warping as mentioned in [16]. The residual between the original frame and the predicted frame is encoded and decoded by a residual autoencoder (denoted as Residual AE in Figure 2c). Finally, the reconstructed frame is the sum of the predicted frame and the reconstructed residual.

F. LOSS FUNCTION

Following the previous deep neural video compression methods, We define a simple rate-distortion loss function, which maximizes the quality reconstruction in terms of PSNR on RGB color space and minimizes the bitrate of the quantized latent tensors, denoted as Z variables with various subscripts for motion and residuals as shown in Figure 2. A scalar parameter is used to balance between the reconstruction quality and the bit-rate, called as the rate-distortion trade-off parameter, denoted as λ_0. The subscript 0 means the base parameter among the hierarchical levels where the rate-distortions for other types of coding frames are derived from λ_0. With GoP is $I_0B_3B_2B_4P_1$ as shown in Figure 1b, the corresponding lambda values are shown in Table 1. In our experiments, we use $\lambda_1 = 0.8\ \lambda_0$, and $\lambda_2 = 0.6\ \lambda_0$. During the training, the loss function is composed of the distortion term, denoted as D_f and the bitrate term, denoted as R_f: The subscript f means for each frame in GoP because we use a hierarchical coding, not a uniform coding. Consequently, the lambda value for each frame depends on its level at a group of picture coding configuration.

$$Loss = \sum\nolimits_{\mathrm{f\ in\ GoP}} (\lambda_f * D_f + R_f) \tag{4}$$

where D_f refers to the distortion or difference between the original frame and the reconstructed frame. The distortion can be MSE (Mean Square Error) loss for PSNR optimization or MS-SSIM for different visual targets. R_f represents the bitrates used for encoding the quantized latent representations of residuals and motion vectors, both associated with the bits used for encoding their corresponding hyperprior [17].

IV. EXPERIMENTS AND RESULTS

A. DATASET AND TRAINING

The Vimeo-90k [19] is a well-known dataset for training video processing tasks. It consists of nearly 90K 7-frame sequences in RGB format. We evaluate the performances of the proposed network and previous neural video codecs on several popular benchmark datasets: UVG [24] with 7 full-HD (1920 × 1080) video sequences, class B of HEVC [25] with 5 full-HD (1920 × 1080) video sequences, class C of HEVC [25] with 4 video sequences with the resolution of 832 × 480, and class D of HEVC [25] with 4 video sequences with the resolution of 416 × 240, all available in raw YUV420 format with various frame rate from 24 fps to 120 fps. For training, firstly a video frame interpolation network based on [18] IFRNet is trained end-to-end with a batch size of 8, crop size 224 × 224 patches from the training samples compressed using JPEG. After obtaining the IFRNet model for video frame interpolation network, we start training the proposed end-to-end deep neural video compression network on original training samples with all types of codecs, I-frame network, P-frame network, and hierarchical B-frame network. We trained the network on 5-frame sequences (from image 1 to image 5 of 7-frame sequences), with the training GoP structure being IBBBP. In the training step, we set the learning rate to 10^{-4}, and randomly extracted 256 × 256 patches. We trained models at four rate-distortion trade-offs, with lambda values equal to $\{0.5, 1, 3, 5\} * 10^{-2}$. For the 100 beginning epoch, we freeze the weight of IFRNet to generate better prediction frame, then we unfreeze it and make the whole network update together to the end of training. The training step took about 3 days on a Nvidia A100 GPU.

B. EVALUATION AND COMPARISONS

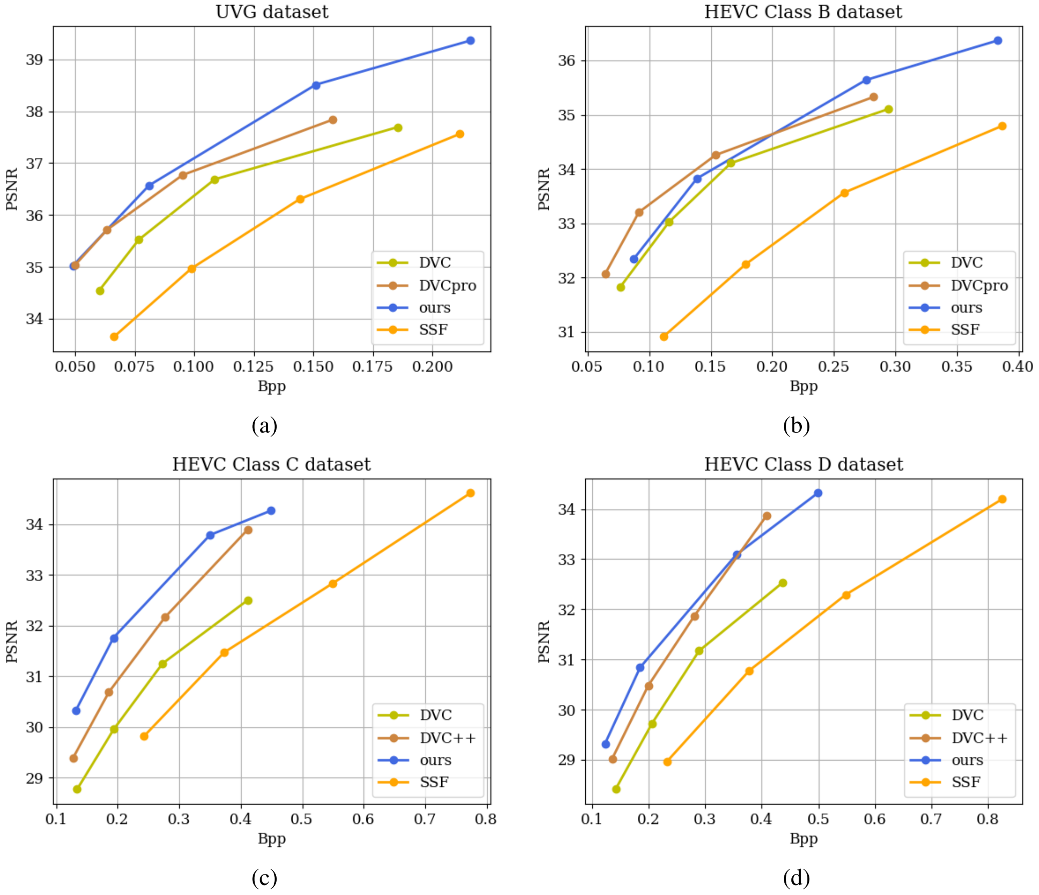
Firstly, we compare our neural video compression model with previous neural video compression models including DVC [13], and DVCPro [14], as well as the baseline, SSF [16] that we use for I-frame and P-frame coding. For DVC and DVCPro, we use results reported by [35], for SSF, we apply the pre-trained models provided by CompressAI library [7] for experiments. With our model, we use a GoP size of 16 for evaluation experiments.

1) COMPARISONS WITH THE PREVIOUS METHODS

As shown in Figure 3 in terms of the rate-distortion curve, the higher curve is the better model. In the UVG dataset, and HEVC-class C dataset our model outperforms the previous neural video compression methods with significant marginals. Similarly, in the HEVC-class B dataset, and HEVC-class D our model outperforms DVC, and SSF models, and is competitive with DVCpro.

2) BJ∅NTEGAARD DELTA RATE (BD-RATE)

in this section, we report BD-rate saving (%) [23] for the same PSNR versus the anchor SSF model [16] that are the



(a) (b) (c) (d)

FIGURE 3. Rate-distortion comparison on the UVG, HEVC-ClassB, HEVC-ClassC, and HEVC-ClassD datasets.

baseline for I-frame and P-frame coding in our video codec. Table 2 presents the average BD-rate saving versus the SSF model on the UVG dataset as 48.01%, HEVC-class B dataset as 50.96%, HEVC-class C dataset as 51.68%, HEVC-class D dataset as 50.39%, and the average of four datasets is 50.26%. It means with the same quality of reconstructed frames, our model reduces bitrate twice in comparison with the baseline SSF model. In addition, we also compare to two previous methods, B-EPIC [33], and BoostSSF [15] which also use SSF as a baseline model for I-frame and P-frame. From the results shown in Table 2, among SSF-based methods, ours are the best and outperforms the others in terms of bitrate saving for PSNR with respect to the anchor SSF model.

3) VISUAL COMPARISON

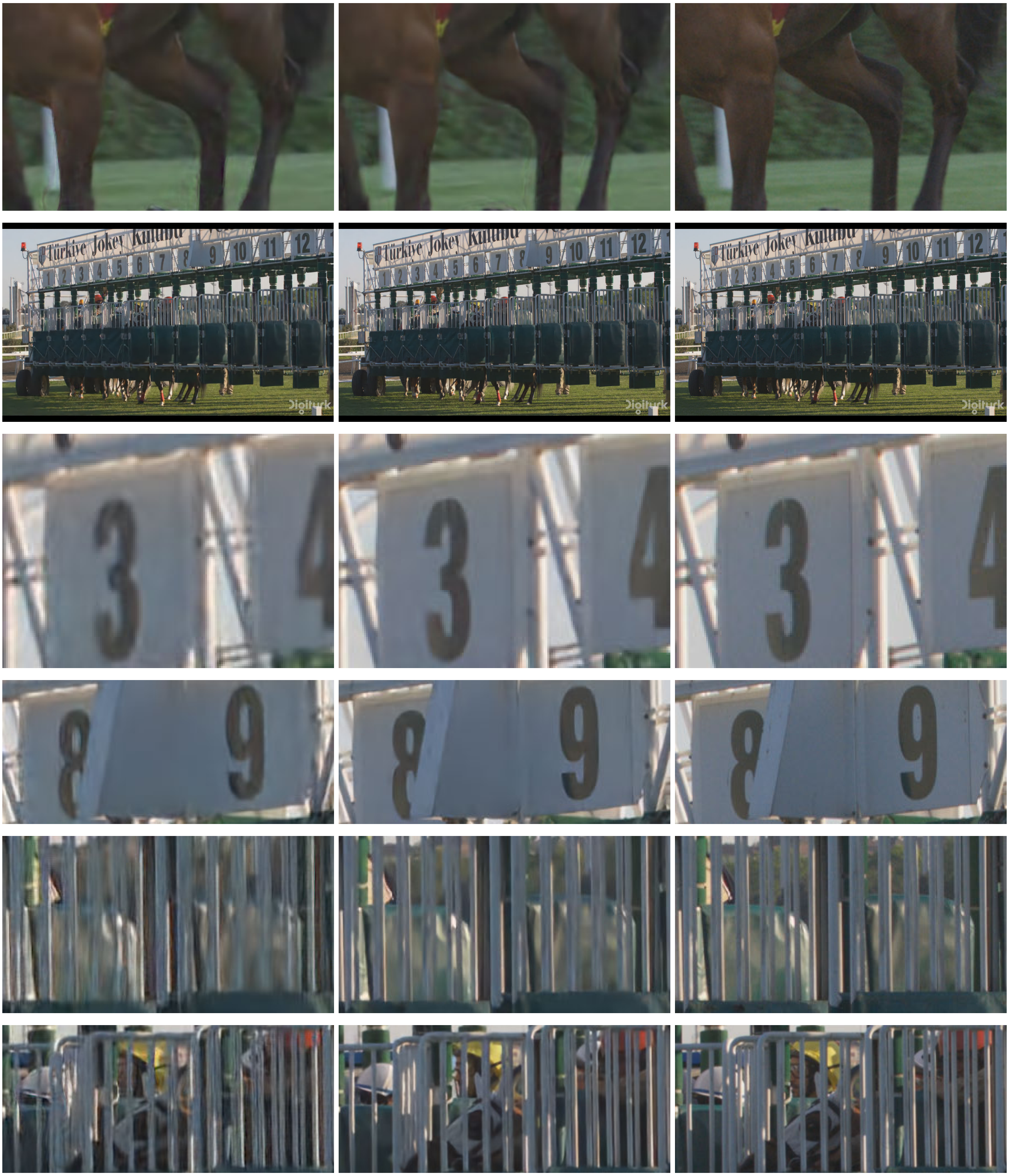
Figure 4 shows the subjective visual comparison between the proposed method and the baseline SSF model [16]. A region of frame 114 of the Jockey sequence and frame 4 of the ReadySetGo sequence of the UVG dataset is shown in the first row and the second row of Figure 4 respectively as comparison examples. Other rows present several regions in the picture (frame 4 of the ReadySetGo sequence) where compression artifacts occurred in the reconstructed frames. As shown in Figure 4, with a similar bitrate, the visual quality of the reconstructed frame generated by our model is better than that of the decoded frame from the SSF model, and our model alleviates the compression artifacts significantly compare to the baseline SSF model.

TABLE 2. Comparisons of BD rate savings for RGB PSNR with respect to the anchor SSF model. Lower is better.

Dataset	UVG	HEVC-B	HEVC-C	HEVC-D	Average
B-EPIC	-28.5%	-19.4%	X	X	-23.95%
BoostSSF	-38.01%	-26.51%	X	X	-32.26%
ours	**-48.01%**	**-50.96%**	**-51.68%**	**-50.39%**	**-50.26%**

4) ABLATION STUDY

We compare the performance between the final model that uses all three reference pictures for B-frame coding, including the intermediate reference frame generated from the video frame interpolation network and two reconstructed reference frames (the previous one and the future one) input for the motion autoencoder, versus the model that only uses the generated intermediate frame. Results shown in Table 3 prove that our final model that applies both unidirectional P-slice



(a) the SSF model (b) the proposed model (c) the ground truth

FIGURE 4. Visual comparison with the anchor SSF model.

coding and bidirectional B-slice coding together via three reference frames is better than the model that uses only B-frame coding via the interpolated reference frame. However, the difference in the contribution to each dataset again verifies the main drawback of a neural video compression model is data dependency.

TABLE 3. BD-Rate (%) comparison for RGB PSNR. The anchor is the final model.

Dataset	UVG	HEVC-B	HEVC-C	HEVC-D	Average
Use only the interpolated frame	15.97%	3.81%	6.16%	5.26%	7.8%

TABLE 4. Running time of our model vs that of the SSF model.

Dataset	resolution	ours (fps)	ssf (fps)
UVG	1920x1080	3.4	3.8
HEVC-B	1920x1080	3.3	3.6
HEVC-C	832x480	11.3	12.7
HEVC-D	416x240	19.2	24.2

5) RUNNING TIME

We report the inference times of our model and compare them with the running times of the baseline model SSF for various resolutions. Experiments are measured on an Nvidia A100 GPU. As shown in Table 4, for the UVG and HEVC-B datasets with full-HD (1920 × 1080) our models can process around 3.4, and 3.3 fps, respectively meanwhile the speed of the SSF model is around 3.8, and 3.6 fps, respectively. For other datasets, please refer to Table 4 for the results. In all datasets, our model is slightly slower than the SFF model, owning to the operations of the frame interpolation network and additional computations. With the significant improvement of coding efficiency around twice, this slight increment of the running time is very acceptable.

6) MODEL COMPLEXITY

We compare the model complexity in model size with respect to the baseline SSF model [16]. The SSF model has 34.2M parameters, and ours has 40.6M parameters. Our model size is larger than that of the SSF model because we add a video frame interpolation network and additional parameters for the motion autoencoder network that use all three reference frames as inputs for the network instead of one reference frame of the SSF model.

V. CONCLUSION

In this paper, a novel framework that integrates a video frame interpolation network to generate an additional reference frame into a hierarchical random access coding of a neural video compression model is presented. The proposed framework not only improves significantly the coding efficiency of the base neural video compression model, bitrate saving 48.01% for the UVG dataset and 50.96% for the HEVC class B dataset respectively but also outperforms the previous methods. The proposed method can be easily extended to other neural video coding models designed for P-frame coding.

REFERENCES

[1] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] B. Bross, Y. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[3] D. Ding, Z. Ma, D. Chen, Q. Chen, Z. Liu, and F. Zhu, "Advances in video compression system using deep neural network: A review and case studies," *Proc. IEEE*, vol. 109, no. 9, pp. 1494–1520, Sep. 2021.

[4] O. Rippel, A. G. Anderson, K. Tatwawadi, S. Nair, C. Lytle, and L. Bourdev, "ELF-VC: Efficient learned flexible-rate video coding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14459–14468.

[5] L. Zhao, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Enhanced motion-compensated video coding with deep virtual reference frame generation," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4832–4844, Oct. 2019.

[6] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5911–5920.

[7] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, "CompressAI: A PyTorch library and evaluation platform for end-to-end compression research," 2020, *arXiv:2011.03029*.

[8] H. M. Cho and K. Choi, "Super-resolution based video coding scheme," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 1777–1779.

[9] M. Khani, V. Sivaraman, and M. Alizadeh, "Efficient video compression via content-adaptive super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4501–4510.

[10] F. Mentzer, G. Toderici, D. Minnen, S. J. Hwang, S. Caelles, M. Lucic, and E. Agustsson, "VCT: A video compression transformer," in *Proc. 36th Conf. Neural Inf. Process. Syst. (NIPS)*, 2022, pp. 1–19.

[11] J. Li, B. Li, and Y. Lu, "Deep contextual video compression," in *Proc. 34th Conf. Neural Inf. Process. Syst. (NIPS)*, 2021, pp. 1–12.

[12] J. Li, B. Li, and Y. Lu, "Hybrid spatial–temporal entropy modelling for neural video compression," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 1503–1511.

[13] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An end-to-end deep video compression framework," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10998–11007.

[14] G. Lu, X. Zhang, W. Ouyang, L. Chen, Z. Gao, and D. Xu, "An end-to-end learning framework for video compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3292–3308, Oct. 2021.

[15] R. Pourreza, H. Le, A. Said, G. Sautière, and A. Wiggers, "Boosting neural video codecs by exploiting hierarchical redundancy," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 5344–5353.

[16] E. Agustsson, D. Minnen, N. Johnston, J. Ballé, S. J. Hwang, and G. Toderici, "Scale-space flow for end-to-end optimized video compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8500–8509.

[17] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–23.

[18] L. Kong, B. Jiang, D. Luo, W. Chu, X. Huang, Y. Tai, C. Wang, and J. Yang, "IFRNet: Intermediate feature refine network for efficient frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1959–1968.

[19] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019.

[20] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, Feb. 1992.

[21] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[22] J. Han, B. Li, D. Mukherjee, C.-H. Chiang, A. Grange, C. Chen, H. Su, S. Parker, S. Deng, U. Joshi, Y. Chen, Y. Wang, P. Wilkins, Y. Xu, and J. Bankoski, "A technical overview of AV1," *Proc. IEEE*, vol. 109, no. 9, pp. 1435–1462, Sep. 2021.

[23] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," Tech. Rep., 2001.

[24] A. Mercat, M. Viitanen, and J. Vanne, "UVG dataset: 50/120 fps 4 K sequences for video codec analysis and development," in *Proc. 11th ACM Multimedia Syst. Conf.*, May 2020, pp. 297–302.

[25] F. Bossen, "Common test conditions and software reference configurations," *JCTVC*, vol. 12, no. 7, 2013.

[26] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 261–270.

[27] H. Lee, T. Kim, T. Chung, D. Pak, Y. Ban, and S. Lee, "AdaCoF: Adaptive collaboration of flows for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5315–5324.

[28] N. Van Thang, K. Lee, and H. Lee, "A stacked deep MEMC network for frame rate up conversion and its application to HEVC," *IEEE Access*, vol. 8, pp. 58310–58321, 2020.

[29] Z. Shi, X. Xu, X. Liu, J. Chen, and M. Yang, "Video frame interpolation transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17461–17470.

[30] L. Lu, R. Wu, H. Lin, J. Lu, and J. Jia, "Video frame interpolation with transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3522–3532.

[31] G. He, C. Wu, L. Xu, L. Li, Z. Xu, W. Xie, and Y. Li, "An efficient video coding system with an adaptive overfitted multi-scale attention network," *IEEE Access*, vol. 9, pp. 64022–64032, 2021.

[32] C. Y. Wu, N. Singhal, and P. Krahenbuhl, "Video compression through image interpolation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 416–431.

[33] R. Pourreza and T. Cohen, "Extending neural P-frame codecs for B-frame coding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6660–6669.

[34] Z. Sun, Z. Tan, X. Sun, F. Zhang, D. Li, Y. Qian, and H. Li, "Spatiotemporal entropy model is all you need for learned video compression," 2021, *arXiv:2104.06083*.

[35] [Online]. Available: https://github.com/ZhihaoHu/PyTorchVideoCompression

NGUYEN VAN THANG received the B.S. degree in electrical engineering from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2010, and the M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2012 and 2019, respectively. He was a Senior Applied Scientist with VinBrain. He is currently a Senior Research Scientist with the AI Camera Center, Viettel High Technology Industries Corporation. He is also an Affiliate Lecturer with the College of Engineering and Computer Science, VinUniversity. His research interests include video processing and computer vision, with a focus on motion analysis, video frame interpolation, and deep neural video compression.

LE VAN BANG received the Ph.D. degree in computer applied technology from the East China University of Science and Technology, in 2018. He is currently the Team Leader of AI computer vision with Viettel High Technology Industries Corporation. He has published multiple articles in top-tier journals. His research interests include computer vision technology, machine learning optimization, and pattern recognition.