Learning-Based Video Coding with Joint Deep Compression and Enhancement

Tiesong Zhao
t.zhao@fzu.edu.cn
[1]Fujian Key Lab for Intell. Process. & Wireless Commun. of Media Info., Fuzhou University
[2]Peng Cheng Laboratory

Weize Feng
201127019@fzu.edu.cn
Fujian Key Lab for Intell. Process. & Wireless Commun. of Media Info., Fuzhou University

Hongji Zeng
201120063@fzu.edu.cn
Fujian Key Lab for Intell. Process. & Wireless Commun. of Media Info., Fuzhou University

Yiwen Xu*
xu_yiwen@fzu.edu.cn
Fujian Key Lab for Intell. Process. & Wireless Commun. of Media Info., Fuzhou University

Yuzhen Niu
yuzhenniu@gmail.com
Fujian Key Lab of Network Comput. & Intell. Info. Process., Fuzhou University

Jiaying Liu
liujiaying@pku.edu.cn
Wangxuan Institute of Computer Technology, Peking University

ABSTRACT

End-to-end learning-based video coding has attracted substantial attentions by compressing video signals as stacked visual features. This paper proposes an end-to-end deep video codec with jointly optimized compression and enhancement modules (JCEVC). First, we propose a dual-path generative adversarial network (DPEG) to reconstruct video details after compression. An α-path and a β-path concurrently reconstruct the structure information and local textures. Second, we reuse the DPEG network in both motion compensation and quality enhancement modules, which are further combined with other necessary modules to formulate our JCEVC framework. Third, we employ a joint training of deep video compression and enhancement that further improves the rate-distortion (RD) performance of compression. Compared with x265 LDP very fast mode, our JCEVC reduces the average bit-per-pixel (bpp) by 39.39%/54.92% at the same PSNR/MS-SSIM, which outperforms the state-of-the-art deep video codecs by a considerable margin. Sourcecode is available at: https://github.com/fwz1021/JCEVC.

CCS CONCEPTS

• **Computing methodologies** → **Image compression**; **Reconstruction**.

KEYWORDS

video coding, deep video compression, end-to-end video codec

ACM Reference Format:
Tiesong Zhao, Weize Feng, Hongji Zeng, Yiwen Xu, Yuzhen Niu, and Jiaying Liu. 2022. Learning-Based Video Coding with Joint Deep Compression and Enhancement. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3548314

*Corresponding author:Yiwen Xu.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548314

1 INTRODUCTION

Figure 1: Reconstructed frames with H.265 (x265 LDP very fast) and JCEVC. The JCEVC achieves a significantly higher RD performance than H.265 by reducing bit-per-pixel (bpp) almost half whilst retaining competitive PSNR or MS-SSIM.

High efficient video compression has been a challenging task in multimedia community since 1980s. Researchers have devoted to improve the rate-distortion (RD) efficiency by introducing more coding tools, such as hierarchical predictions, coding tree units and asymmetric partitions, to hybrid video coding structure. In each generation of video codec, these consisting efforts approximately halves the compressed bits at the same visual quality. Among them, the popular H.265/HEVC [39] and H.266/VVC [7] are considered as the newest achievements of Joint Collaborative Team on Video Coding (JCT-VC). With the widespread use of high definition (HD) videos, it is undoubtable that the video coding problem is still a critical issue in the 5G or B5G era.

Popular video codecs treat the videos as *signals*. They remove the spatial-temporal redundancies of videos by low-level transform, quantization and entropy coding. However, in computer vision community, the videos can be processed as stacked *features*, which allows us to develop end-to-end video codecs with big data and learning. Recently, the learning-based image codecs [8–11, 13, 17, 20, 22, 29, 44] have surpassed the traditional image codecs in terms of

compression efficiency, which also inspired learning-based codecs for videos. In fact, we have witnessed a booming of learning-based video codecs in the past two years.

These learning-based video codecs utilize deep neural networks to imitate motion estimation (ME), motion compensation (MC), residual compression and video reconstruction [3, 15, 18, 19, 21, 24–28, 33, 34, 45, 46]. Owing the advantage of deep learning on large-scale datasets, these methods adopt convolutional neural networks (CNNs), auto-encoders, and/or generative adversarial networks (GANs) to achieve high coding performances. Among them, [45, 46] utilized bi-directional prediction while the other methods used one-way prediction with 1 to 4 references. A separate quality enhancement network was deployed in post-processing stage of [45]. Early works exhibited superior performances than H.264/AVC or competitive with H.265/HEVC [3, 15, 27, 34]. Recently, deep video codecs have surpassed H.265/HEVC [18, 19, 21, 24–26, 28, 33, 45, 46]. In such sense, end-to-end learning-based video codecs have paved another way of video compression.

Despite of these great efforts, it is still imperative to further improve the RD efficiency of deep video coding. In this paper, we move the next step that benefits from GAN-based visual enhancement. To improve the full-reference reconstruction quality, we introduce a parallel path with residual attention blocks (RABs). This dual-path enhancement with GAN (DPEG) network is co-trained by a generative-adversarial process to well reconstruct video frames after quantization, where a convolutional long short-term memory (ConvLSTM) network [37] is also incorporated to refer to multiple coded frames. In our codec, this design is reused in both MC and perceptual enhancement, while the optical flow, auto-encoders and residual networks are utilized to construct the other modules. Aiming at an optimal RD performance, we employ a joint training of modules. The proposed joint compression and enhancement for deep video coding (JCEVC) achieves superior performance than H.265, as depicted in Figure 1.

Our main contributions are summarized as follows.

A DPEG network for video reconstruction: We propose the DPEG with two paths of different receptive fields. An α-path focuses on structure features with auto-encoder and ConvLSTM. A β-path focuses on texture details with RABs. The fusion of these paths improves the RD efficiency of deep coding.

A JCEVC framework for end-to-end deep video coding: We propose the JCEVC framework by reusing DPEG network in both MC and perceptual enhancement. The other modules of our JCEVC are constructed by CNN-based optical flow, auto-encoder and residual networks.

Joint training of video compression and enhancement: We employ a joint training of compression and enhancement to achieve an optimal RD tradeoff. To the best of our knowledge, we are the first to jointly optimize compression and enhancement in end-to-end deep video coding. Experimental results reveal the effectiveness of our JCEVC with joint training.

2 RELATED WORK

Deep image compression. The reigning image codecs, such as JPEG [41], JPEG2000 [40] and BPG [2], employ frequency transform and quantization to remove spatial redundancies of images. While in deep image codecs, auto-encoders, recurrent neural networks (RNNs) and GANs are widely used. Recent efforts have supported spatial rate allocation, *i.e.*, to allocate bits based on spatial textures and contexts [8, 11, 13, 20, 22] or multiple bpps with one network [9, 44]. In [10], CNN-based ProxIQA was proposed to mimic the perceptual model for RD tradeoff. In [13], discretized Gaussian mixture likelihoods were utilized to parameterize latent code distributions, aiming at a more accurate and flexible entropy model. In [17], a checkerboard context model was proposed to support parallel image decoding. In [29], the CNN was employed to design a wavelet-like transform for removing redundancies. These methods exploits spatial correlations of single pictures, while our JCEVC framework focuses on spatial-temporal correlations of successive pictures, especially the MC module based on DPEG.

Deep video compression. By removing the spatial-temporal redundancies of video frames,the reigning video codecs, such as H.265/HEVC and H.266/VVC, achieve significantly higher compression efficiency than image codecs. These characteristics have also been utilized in deep video coding. A classic method, called deep video compression (DVC) [27], replicated ME/MC, transform /quantization and entropy coding with optical flow, non-linear residual encoder and CNN, respectively. In [26], an error-propagation-aware training was proposed to address error propagation and content adaptive compression in DVC. In [28], two variants of DVC, DVC Lite and DVC Pro, were designed with different coding complexities. In [24], CNN-based motion vector (MV) prediction, MV refinement, multi-frame MC and residual refinement were introduced to develop multi-frame prediction for learned video compression (M-LVC). In [45], hierarchical learned video compression (HLVC) introduced hierarchical group-of-picture (GOP) structure which had shown its high efficiency since H.264 scalable coding. It also utilized a weighted recurrent quality enhancement to further improve the visual quality at decoder-end. In [46], a recurrent learned video compression (RLVC) employed recurrent auto-encoder and recurrent probability model for improved MV and residual compression.

The rate allocation of deep video coding was first realized by [15] and [34], which utilized deep generative model and recursive network for high compression efficiency. [3] designed a scale-space flow to improve ME robustness under common failure cases, *e.g.*, dis-occlusion and fast motion. [18] designed a resolution-adaptive flow coding (RaFC) to effectively compress optical flow maps globally and locally. In [21], encoder complexity is optimized with less model parameters. In [33], an efficient, learned and flexible video coding (ELF-VC) was proposed to support flexible rate coding with high RD efficiency. Recently, [19] and [25] came up with different ways (CNN or GAN) to compress video contents via low-dimensional feature representations, which were further utilized to reconstruct video frames. In our work, the dual-path DPEG network is introduced to extensively reduce the spatial-temporal redundancies and show a significantly increased compression efficiency.

Visual enhancement. The visual enhancement techniques are utilized to improve the visual quality of images or videos. Generally, image enhancement can be achieved by GAN [12, 31] or CNN [23, 38]. In [47], a multi-frame quality enhancement (MFQE) method utilized high-quality frames to enhance low-quality frames, where the high-quality frames were detected with support vector machine (SVM). In [14], MFQE2.0 method replaced the SVM by bi-directional

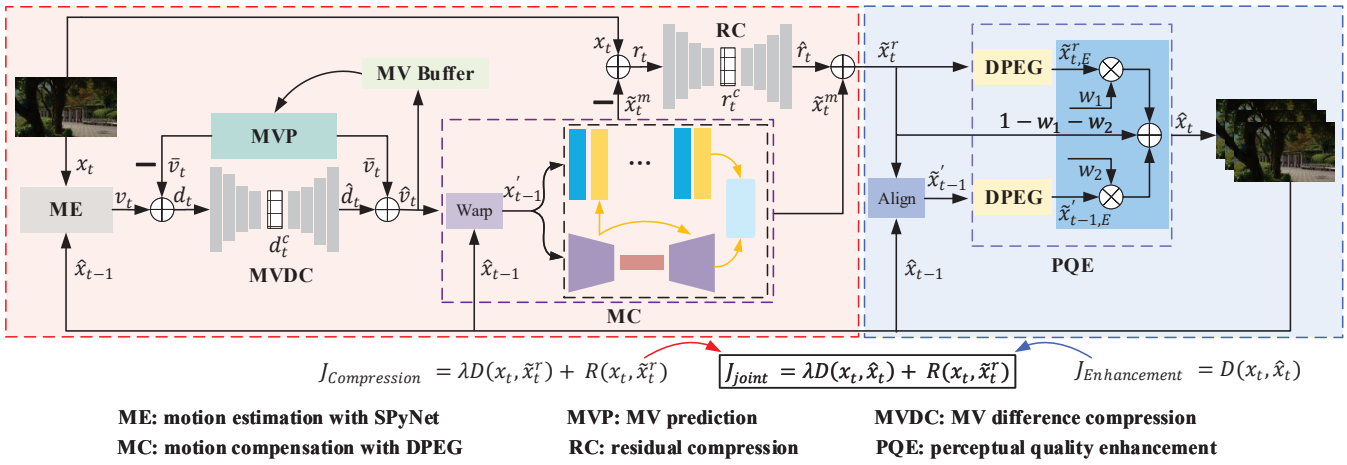

Figure 2: The framework of our JCEVC encoder, where both MC and PQE modules are realized by our DPEG network. For a frame x_t, the ME module calculates its MV v_t with SPyNet and bi-directional IPPP structure. Meanwhile, the MVP module derives an MV prediction, $\bar{v}_t$, based on an MV buffer of references. The MVDC module compresses and reconstructs the MV difference $d_t = v_t - \bar{v}_t$ with an auto-encoder, quantization and entropy coding. Then, the MC module utilizes the reconstructed MV, $\hat{v}_t = \bar{v}_t + \hat{d}_t$, to align the reconstructed frame $\hat{x}_{t-1}$ to x_t. The warped frame x'_{t-1} is enhanced by the DPEG network to generate a compensated frame $\tilde{x}_t^m$. After that, the RC module compresses and reconstructs the texture residual $r_t = x_t - \tilde{x}_t^m$. Finally, the PQE module receives the residual compensation $\tilde{x}_t^r = \tilde{x}_t^m + \hat{r}_t$ and the aligned reference $\tilde{x}'_{t-1}$, and employs two DPEGs with weighted fusion to obtain the reconstructed frame $\hat{x}_t$. Aiming at an optimal RD efficiency, the compression and enhancement processes are jointly trained.

long short-term memory (BiLSTM) and also improved the convolutional network of MFQE. [43] proposed a task-oriented flow (TOFlow) with superiority to optical flow in video enhancement. [16] leveraged spatial-temporal relationship of videos and proposed to simultaneously increase their spatial resolutions and frame rates. The visual enhancement is also introduced at the decoder-end of HLVC [45]. In our work, we incorporate the enhancement module into the reconstruction process, thereby leading to a joint training of video compression and enhancement.

3 PROPOSED METHOD

Notations. Let x_t denote the t-th frame in picture coding order of an original video and $\hat{x}_t$ denote its constructed frame. $\tilde{x}_t^m$ and $\tilde{x}_t^r$ represent the compensated frames of x_t with motion and residual information, respectively. The reconstructed $(t-1)$-th frame, $\hat{x}_{t-1}$, is sent back for recurrent prediction. It is also aligned to x_t and $\tilde{x}_t^r$, resulting to x'_{t-1} and $\tilde{x}'_{t-1}$, for MC and quality enhancement. The MV matrix of x_t is predicted as $\bar{v}_t$ and finally denoted as v_t after ME. Their difference, $d_t = v_t - \bar{v}_t$, is compressed and reconstructed as $\hat{d}_t$. The difference between x_t and $\tilde{x}_t^m$ is represented as a residual r_t, whose corresponding reconstruction is denoted by $\hat{r}_t$.

3.1 The JCEVC framework

As shown in Figure 2, the encoder of JCEVC consists of 6 modules: ME, MVP, MVDC, MC, RC and PQE, among which we redesign the MC and PQE modules and improve the remaining modules. All modules are jointly trained for an optimized RD performance. This coding procedure applies to all P frames while the context-adaptive entropy model of [22] is utilized to compress I frames.

ME module. In JCEVC, we employ a bi-directional IPPP structure [46] with a GOP size of 15. The frames 0, 15 are coded as I frames while the others are coded as P frames. Among them, the frames 1~7 are forwardly predicted from frame 0 while the frames 8~14 are backwardly predicted from frame 15 of next GOP. To remove temporal redundancies, the ME module estimates the MV v_t between adjacent frames: x_t and $\hat{x}_{t-1}$. In this paper, we employ a low-complexity optical flow model, SPyNet [32], which combines spatial pyramid and deep convolutional network for fast ME.

MVP module. Due to high spatial-temporal correlations between MVs, it is sensible to compress the MV difference d_t instead of MV values. We set $d_t = v_t - \bar{v}_t$, where $\bar{v}_t$ is a predicted MV from an MV buffer, which is constituted by the MV information of its three preceding frames. The prediction is achieved by a light network with a convolutional layer, two residual blocks and another two convolutional layers. The channel number is 2 for the last layer and 64 for each of the others. The convolutional kernel size and stride are 3×3 and 1, respectively. Relu is utilized as the activation function in all convolutional layers.

MVDC module. The MV difference $d_t \in R^{H\times W\times 2}$, where H and W are frame height and width, is compressed by MVDC module. To reduce the coding bits, we employ an auto-encoder with four downsampling layers and four upsampling layers that are implemented with convolutions and deconvolutions, respectively. The compact representation $d_t^c \in R^{\frac{H\times W}{16}\times 128}$ is further processed by quantization and entropy coding, with the procedure presented in [27]. The reconstructed MV can be calculated as $\hat{v}_t = \bar{v}_t + \hat{d}_t$.

MC module. The MC module utilizes the reconstructed previous frame, $\hat{x}_{t-1}$ and the reconstructed MV, $\hat{v}_t$, to generate a warped

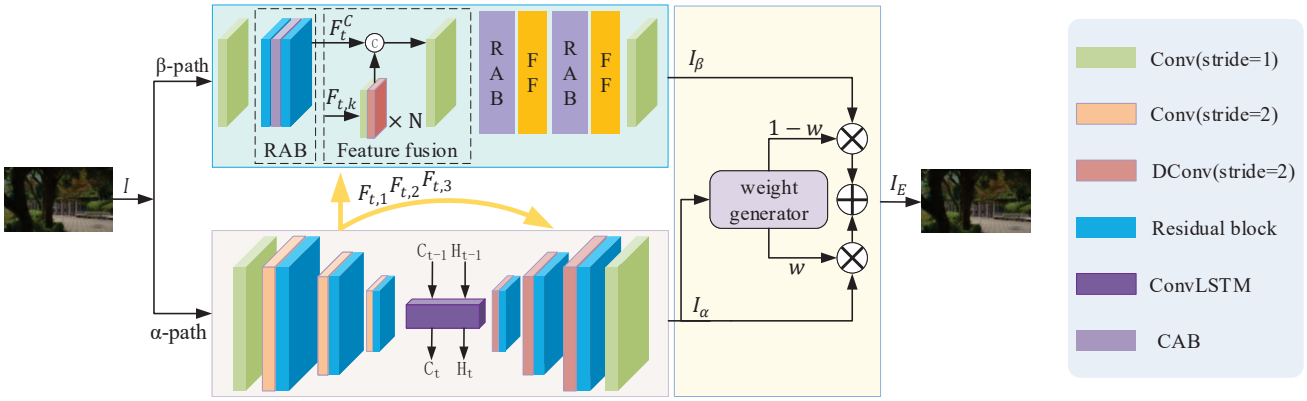

Figure 3: The proposed DPEG network. An α-path with generator and ConvLSTM focuses on larger receptive field and global structures; a β-path with RABs focuses on smaller receptive field and local textures. Their outputs are combined with a weighted fusion. Sematic features ($F_{t,1}, F_{t,2}, F_{t,3}$) are fed from α-path into β-path as a guidance. Here I and I_E are task-dependent, *e.g.* $I = x'_{t-1}, I_E = \tilde{x}_t^m$ in MC.

frame that is aligned to the current frame x_t. The warped frame, namely x'_{t-1}, is fed into the DPEG network to reconstruct an enhanced frame $\tilde{x}_t^m$. A ConvLSTM model is deployed to use multiple reference frames in history. This module will be elaborated in Section 3.2.

RC module. The motion compensated and enhanced frame $\tilde{x}_t^m$ is further utilized to calculate the texture residual $r_t = x_t - \tilde{x}_t^m$. Its compression is finished with the same process to that of MDVC. After this step, the reconstructed texture residual and compensated frame are represented by $\hat{r}_t$ and $\tilde{x}_t^r = \tilde{x}_t^m + \hat{r}_t$, respectively.

PQE module. The last module employs two DPEGs to enhance $\tilde{x}_t^r$ and $\tilde{x}'_{t-1}$, and further combines them as the reconstructed frame $\hat{x}_t$. In particular, $\tilde{x}'_{t-1}$ is a warped frame that aligns $\hat{x}_{t-1}$ to $\tilde{x}_t^m$, where the alignment process is realized by an ME and a warping operation. Details of this module will be presented in Section 3.3.

The decoder. The compressed stream of JCEVC consists of compressed MV residuals, compressed texture residuals and the information of previously coded frames, from which we can easily obtain $\hat{d}_t$, $\hat{r}_t$, $\hat{x}_{t-1}$ and $\overline{v}_t$. Then $\tilde{x}_t^m$ can be derived with $\hat{v}_t = \overline{v}_t + \hat{d}_t$, $\hat{x}_{t-1}$ and the MC module. Finally, the reconstructed frame $\hat{x}_t$ is obtained by $\tilde{x}_t^r = \tilde{x}_t^m + \hat{r}_t$, $\hat{x}_{t-1}$ and the PQE module.

3.2 MC with DPEG network

In video coding, P frames generally have lower residuals than I frames. The residual at t-th frame, r_t is calculated between the current frame x_t and its motion compensated reference $\tilde{x}_t^m$. To further reduce the bits for r_t under the same visual quality, we first compensate the previous frame with a non-linear warp,

$$x'_{t-1} = \text{Warp}(\hat{x}_{t-1}, \hat{v}_t); \tag{1}$$

and then enhance the warped frame with DPEG network,

$$\tilde{x}_t^m = \text{DPEG}(x'_{t-1}). \tag{2}$$

The proposed DPEG network is implemented with dual-path GAN and RABs. To date, dual-path networks have been adopted in parallel processing of image denoising and enhancement tasks; however, they have not yet been exploited in image reconstruction after compression. The basic generator of GAN involves a de facto downsampling process to increase its receptive field, whilst excluding texture details in the original frame. This is unhelpful in frame reconstruction that is evaluated by full-reference quality metrics. To address this issue, we add a complementary path with RABs for video details, as shown in Figure 3. To avoid high computational complexity, the DPEG is designed as a compact framework with input I and output I_E. In the second step of MC as Equation (2), $I = x'_{t-1}, I_E = \tilde{x}_t^m$.

α-path. To increase the receptive field of low-dimensional features, an encoder is employed with a convolutional layer with stride 1, three convolutional layers with stride 2 and three residual blocks. The obtained sematic features, $F_{t,1} \in R^{H\times W\times C}$, $F_{t,2} \in R^{\frac{H}{2}\times\frac{W}{2}\times C}$, $F_{t,3} \in R^{\frac{H}{4}\times\frac{W}{4}\times 2C}$, are fed into β-path and the decoder. After that, a ConvLSTM is inserted to fully utilize the reference information of coded frames. The state and output vectors of ConvLSTM, C_{t-1} and H_{t-1}, are fed into current network. The decoder part is set as an inverse process of encoder with upsampling. To avoid sematic information loss, a U-Net [35] is used to skip connect the sematic features to the decoder so that the enhanced results contain the original sematic information.

β-path. This path consists of two convolutional layers with stride 1, three RABs and three feature fusion blocks where downsamlping is not applied. An RAB is composed of two residual blocks and a channel attention block (CAB) [48]. Each RAB extracts a feature representation $F_t^C \in R^{H\times W\times C}$ of C channels, which is further fused with the sematic feature $F_{t,k}, k = 1, 2, 3$ from α-path. The feature fusion block also includes an upsampling process to match the dimensions of features. With a smaller receptive field, there is less access to global features in β-path. Therefore, the fusion of sematic features from α-path benefits the frame reconstruction.

Weighted fusion. To take advantage of both paths, we employ a weighted summation of results:

$$I_E = w \cdot I_\alpha + (1 - w) \cdot I_\beta, \tag{3}$$

where w is a weight matrix to represent the dependence degree of I_E on I_α. We utilize three convolutional layers and two residual blocks to extract the saliency of frame and further warp it to $(0, 1)$ with a Sigmoid function.

The discriminator. The two paths of DPEG are co-trained by a generative-adversarial process after the weighted fusion of Equation (3). The training of this process involves a discriminator with attention mechanism. First, it utilizes four downsampling layers to achieve a larger receptive field. Then, it employs attention mechanism and different pooling strategies to generate two attention maps, M_{avg} and M_{max}. After that, it multiplies the downsampled frame with the two attention maps and concatenates them as an feature map. Finally, it judges the obtained feature map after two convolutions with stride 1. The M_{avg} and M_{max} are also important in the loss function for training.

3.3 PQE with DPEG networks

The compensated video frames are further enhanced before reconstruction. There have been extensive studies to enhance the visual quality or remove compression artifacts of video sequences. The quality enhancement module has also been introduced to decoder-end of deep video compression by [45]. In this paper, we deploy a PQE module before reconstruction, which is thus jointly trained by the encoder. The DPEG network is reused in this module due to its effectiveness in visual enhancement.

Aiming at a better visual quality, two parallel DPEG networks are adopted to enhance the compensated frame $\tilde{x}_t^r$ and the aligned previous frame $\tilde{x}_{t-1}^{'}$, respectively. The $\tilde{x}_t^r$ is compensated by $\tilde{x}_t^m$ and the reconstructed residual $\hat{r}_t$. The $\tilde{x}_{t-1}^{'}$ represents the results of aligning $\hat{x}_{t-1}$ to $\tilde{x}_t^r$, where the alignment process is a combination of ME and warping: an MV is firstly calculated by SPyNet and then utilized to warp $\hat{x}_{t-1}$:

$$\tilde{x}_{t-1}^{'} = \mathrm{Warp}(\hat{x}_{t-1}, \mathrm{ME}(\hat{x}_{t-1}, \tilde{x}_t^r)). \tag{4}$$

The $\tilde{x}_t^r$ and $\tilde{x}_{t-1}^{'}$ frames are separately enhanced as $\tilde{x}_{t,E}^r$ and $\tilde{x}_{t-1,E}^{'}$, which are fused to obtain the final reconstruction $\hat{x}_t$:

$$\hat{x}_t = w_1 \cdot \tilde{x}_{t,E}^r + w_2 \cdot \tilde{x}_{t-1,E}^{'} + (1 - w_1 - w_2) \cdot \tilde{x}_t^r, \tag{5}$$

where the weights w_1 and w_2 are generated with the weight generator in Section 3.2 but are halved to avoid data overflow after summation. The weight $1 - w_1 - w_2$ is a penalty coefficient in case of enhancement failures.

3.4 Joint training of JCEVC

Our training process consists of two phases. In the first phase (0 ~ 300K iterations), we perform a coarse-grain training with the reference frame x_{t-1} and learning rate 1e-4; while in the second phase (300K ~ 900 K iterations), we perform a fine-grain training with the reference frame $\hat{x}_{t-1}$ and learning rate 1e-5. In each phase, we successively train the MVP, MVDC, MC, RC, PQE modules and then perform a joint training of all modules. The ME module is performed with the SPyNet without further training.

The reasons to adopt this joint training strategy are as follows. In reigning video codecs, the RD optimization theory was proposed to minimize the RD cost of

$$J_{\text{Compression}} = \lambda D(x_t, \tilde{x}_t^r) + R(x_t, \tilde{x}_t^r), \tag{6}$$

where D and R refer the compression distortion and bitrate, separately. The objective of enhancement is to further reduce the distortion

$$J_{\text{Enhancement}} = D(x_t, \hat{x}_t) = D(x_t, \mathrm{PQE}(\tilde{x}_t^r, \hat{x}_{t-1})). \tag{7}$$

Compared with reigning video codecs, the deep video codec has an advantage that its end-to-end framework can be jointly optimized with training on a large-scale dataset. Taking this advantage, we propose to jointly train the compression and enhancement modules of deep video coding. The objective is then set as to minimize the total RD cost

$$J_{\text{joint}} = \lambda D(x_t, \hat{x}_t) + R(x_t, \tilde{x}_t^r). \tag{8}$$

Obviously, the joint training releases the burden of reconstruction. The compression stage allows a higher D, which can be eliminated in the enhancement stage, with a reduced R. Hence, the overall RD tradeoff is improved.

Inspired by the RD cost function, we employ the following loss functions for the MVP, MVDC, MC, RC, PQE and joint training:

$$\begin{cases} \mathcal{L}_{\text{MVP}} = \mathrm{MSE}(v_t, \overline{v}_t) \\ \mathcal{L}_{\text{MVDC}} = \lambda \mathrm{MSE}(d_t, \hat{d}_t) + R_{mvd} \\ \mathcal{L}_{\text{MC}} = \mathcal{L}_{\mathcal{G}} \\ \mathcal{L}_{\text{RC}} = \lambda \mathrm{MSE}(r_t, \hat{r}_t) + R_{res} \\ \mathcal{L}_{\text{PQE}} = D(x_t, \hat{x}_t) \\ \mathcal{L}_{\text{ALL}} = \lambda D(x_t, \hat{x}_t) + (R_{mvd} + R_{res}) \end{cases}, \tag{9}$$

where MSE denotes the mean squared error. R_{mvd} and R_{res} represent the bits consumed by MV difference and residuals after compression, respectively. $D(\cdot)$ represents the frame-level distortion, which is calculated as MSE and 1−MS-SSIM in PSNR-oriented and MS-SSIM-oriented codecs, respectively. λ is a coefficient for RD tradeoff. $\mathcal{L}_{\mathcal{G}}$ is the loss function for generator of DPEG. The loss functions for generator and discriminator are set as:

$$\begin{aligned} \mathcal{L}_{\mathcal{G}} = \gamma & E_{x\sim\hat{x}}[(x_t - \mathcal{G}(x_{t-1}^{'}))^2] \\ & + E_{x\sim\hat{x}}[1 - \mathcal{D}(\mathcal{G}(x_{t-1}^{'}))^2] \\ & + E_{x\sim\hat{x}}[1 - \phi(\mathcal{G}(x_{t-1}^{'}))^2], \end{aligned} \tag{10}$$

$$\mathcal{L}_{\mathcal{D}} = E_{x\sim\hat{x}}[1 - \mathcal{D}(x_t)^2] + E_{x\sim\hat{x}}[\mathcal{D}(\mathcal{G}(x_{t-1}^{'}))^2], \tag{11}$$

where $\phi(\cdot)$ represents the calculation of attention maps M_{avg} and M_{max}.

4 EXPERIMENTS

4.1 Experimental setup

Datasets. We train our JCEVC codec with the popular Vimeo-90k [43] dataset, which consists of 89,000 video clips at a resolution of 448×256. To report the performance of our method, we test on H.265 CTC (including Class B at 1920×1080, Class C at 832×480 and Class D at 416×240) [6], MCL-JCV (at 1920×1080) [42], UVG (at 1920×1080) [30] and VTL (at 352×288) [1]. In total, there are 42 HD videos and 23 low-resolution videos are tested.

Evaluation. We compare our method with the popular deep codecs FVC [19], Liu's [25], RLVC [46], Agustsson's [3], HLVC [45], M-LVC [24], Hu's [18], Lu's [26], DVC [27] as well as H.265 implemented by x265 LDP very fast mode. For fair comparison, the results of compared methods are collected from their reports. The consumed bits and reconstruction quality are evaluated by bpp and

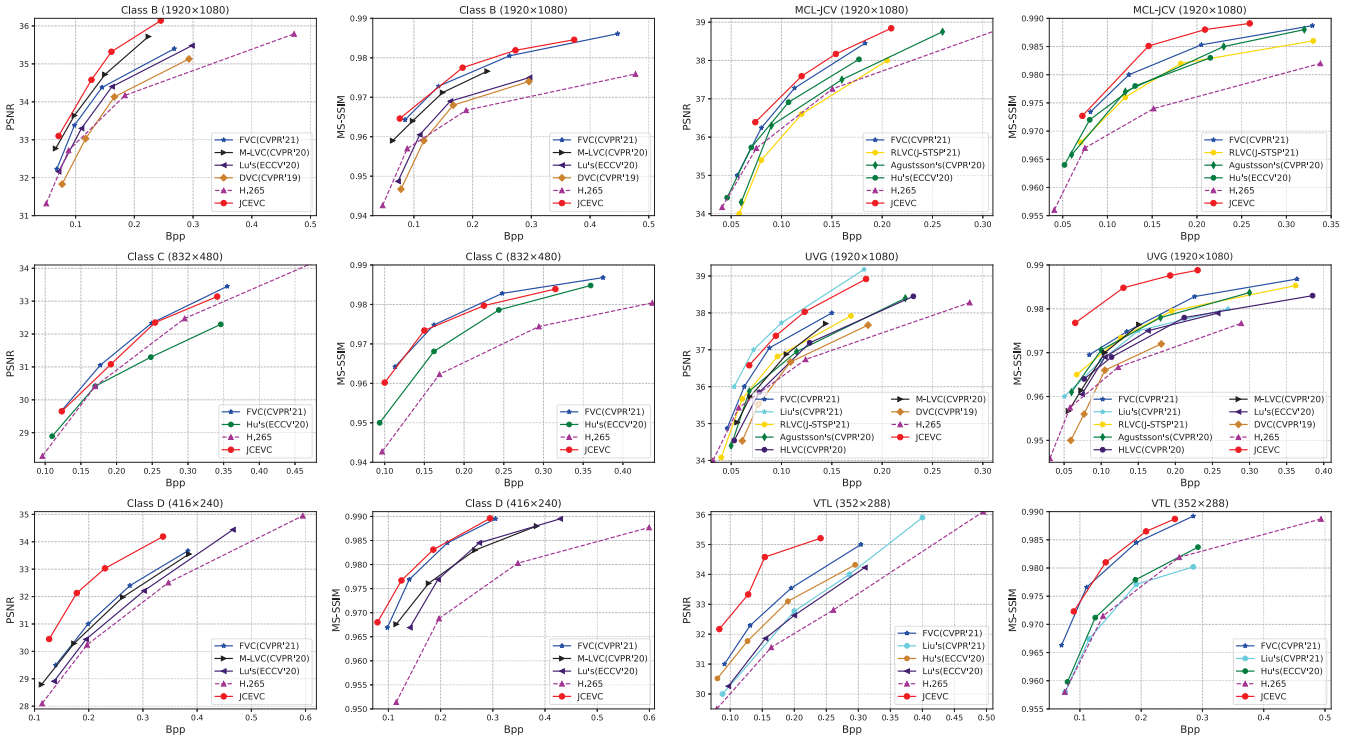

Figure 4: Compression of our JCEVC with 9 popular deep codecs and H.265 (x265 LDP very fast). The proposed JCEVC exhibits significantly superior or at least competitive performance compared with the state-of-the-arts in each dataset.

Datasets	DVC [27]	Hu's [18]	Lu's [26]	Agustsson's [3]	HLVC [45]	M-LVC [24]	RLVC [46]	Liu's [25]	FVC [19]	JCEVC
Class B	5.66/-2.74	–/–	-13.35/-7.93	–/–	-11.75/-37.44	-36.55/-42.82	-24.20/-50.42	–/–	-23.75/ -54.51	-44.19 /-58.29
Class C	25.88/-6.88	4.94/-32.44	–/–	–/–	7.83/-23.63	–/–	-4.67/-35.94	–/–	-14.18 / -43.58	-8.58 /-44.10
Class D	15.34/-18.51	–/-32.43	-6.86/–	–/–	-12.57/ -52.56	-13.87/-36.27	-27.01 /-48.85	–/–	-18.39/-51.19	-44.72 /-56.38
MCL-JCV	–/–	-10.60/-34.10	4.21/–	-1.82/-33.61	–/–	–/–	–/–	–/–	-22.48 /-52.00	-31.21 / -51.17
UVG	10.40/8.05	–/–	-7.56/-25.49	-8.80/-38.04	-1.37/-30.12	-12.11/-25.44	-13.48/-40.62	-49.42 /-30.70	-28.71/ -45.25	-47.62 / -77.60
VTL	–/–	–/-6.04	-16.05/–	–/–	–/–	–/–	–/–	-9.51/2.42	-28.10 / -39.44	-60.02 /-41.98
Average	**8.03/-5.02**	**-2.83/-26.25**	**-7.92/-16.71**	**-5.31/-35.83**	**-4.47/-35.94**	**-20.84/-34.84**	**-17.34/-43.96**	**-29.4 /-14.14**	**-22.60/ -46.66**	**-39.39 /-54.92**

Table 1: Compression on BDBR results calculated by the PSNR vs. bpp and MS-SSIM vs. bpp curves. H.265 is set as the benchmark. On average, our JCEVC significantly outperforms the state-of-the-arts.

PSNR/MS-SSIM, respectively. We also calculate the BDBR values [4] that represents the average bit reduction with the same PSNR or MS-SSIM.

Implementation details. We implement our model on Tensorflow with all training and testing performed on an NVIDIA RTX 2080Ti GPU. The batch size and γ are set as 4 and 1000, respectively. For PSNR-oriented compression, we train four models with different λ values from 512 to 2560; while for MS-SSIM-oriented compression, we train another four models with λ from 8 to 48. Detailed training process can be seen in Section 3.4.

4.2 Experimental results

Figure 4 shows the comparison between our JCEVC and the state-of-the-arts. To evaluate our method to the maximum extent, we test 6 video groups from 4 datasets and collect all available results of compared codecs. The performances of codecs are shown by two types of curves: PSNR vs. bpp and MS-SSIM vs. bpp. A curve above others is considered with a superior RD performance. Three conclusions can be drawn from the figure. **First**, all deep codecs achieve competitive or superior performances compared with H.265 (x265 LDP very fast), which demonstrates the effectiveness of learning-based video coding. **Second**, the deep video coding has been greatly improved since 2019, which demonstrates the potential of learning-based video coding. The recent deep codecs, such as FVC and Liu's, have significantly surpassed the H.265. We can envision a deep video codec with comparable performance to H.266/VVC in the foreseeable future. **Third**, our JCEVC achieves significantly superior performances than the state-of-the-arts in most datasets. For example, in Class D by PSNR, UVG by MS-SSIM and VTL by PSNR, the JCEVC achieves remarkably higher performance even compared

with the 2nd best curves. In Class C by PSNR and MS-SSIM, the JCEVC achieves comparable performances to FVC. While in other figures, the JCEVC also surpasses all compared methods. These facts undoubtedly demonstrate the superiority of our JCEVC model.

To quantitatively compare the video codecs, we also present the BDBR results (with PSNR and MS-SSIM) of all available curves in Table 1, where H.265 is also set as the benchmark. These results are consistent with those in Figure 4 that our JCEVC always ranks the best or 2nd best in all datasets. An interesting result occurs when comparing FVC with JCEVC in MCL-JCV by MS-SSIM. The RD curves indicates JCEVC is superior while the BDBR slightly prefers the FVC. This conflict is due to the different definition domains to interpolate and calculate BDBR [4], which is unusual and does not affect the conclusion. On average, the JCEVC achieves a BDBR of -39.39% or -54.92% by PSNR or MS-SSIM. This fact also supports the superior efficiency of the JCEVC.

By summarizing Figure 4 and Table 1, there are some minor issues to be clarified. **First**, all codecs show weak improvements in CTC Class C. This fact might be attributed to higher motions in this Class [26]. In particular, the temporal information (TI) values of Classes B, C, D are 18.6, 24.0 and 21.5, respectively. Despite that, our JCEVC still ranks the best and 2nd best in terms of BDBR by PSNR and MS-SSIM, respectively. **Second**, fair comparison with visual enhancement. This module was also adopted in HLVC. As shown above, our JCEVC is significantly superior to this method in terms of BDBR. **Third**, fair comparison with bi-directional IPPP structure. This structure was also used by RLVC while the hierarchical B structure was introduced by HLVC. Our JCEVC significantly outperforms the above two methods in terms of BDBR.

Regarding to computational complexity, some existing codecs did not report their time costs. Among all available codecs (DVC [27], Lu's [26], M-LVC [24], RLVC [46], FVC [19] and JCEVC, where the RLVC includes entropy coding for fair comparison), the time cost magnitudes are in the order of 1e-2s to 1s per frame. Our JCEVC is with a medium complexity of 0.246s (resp. 0.224s) per frame to encode (resp. decode) Class D on 1080Ti. Its model parameter size is 14.9M. This complexity is acceptable considering its promisingly high RD improvement compared with its peers.

4.3 Ablation study

The ablation experiments are conducted with PSNR-based JCEVC. Similar conclusions can be drawn with MS-SSIM.

Contributions of all modules. In JCEVC, we design a light MVP, an MC with DPEG, a PQE with DPEGs and a joint training, as shown in Figure 2. To examine the contributions of these modules, we perform the following ablation study. A baseline method with a simple but feasible framework (ME+MVDC+MC w/o DPEG+RC) is examined first. Then, our designed modules (MVP, MC w/ DPEG, PQE, joint training) are introduced sequentially to observe their RD improvements. The average results on CTC class D are summarized in Figure 5. With more designed modules, the RD performance is continuously improved. For example, with PSNR=32dB, the bpps of the five settings are 0.47, 0.45, 0.28, 0.20, 0.17, respectively, which indicate the bpp savings at 4.3%, 37.8%, 28.6% and 15.0% by introducing MVP, MC w/ DPEG, PQE and joint training. This fact reveals the

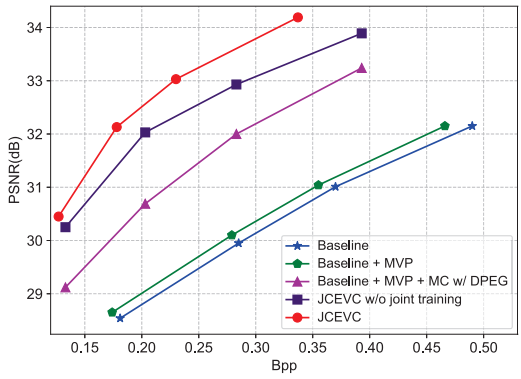

Figure 5: Ablation study of all JCEVC modules. The RD performances are kept improved with more modules, which demonstrates the effectiveness of our design.

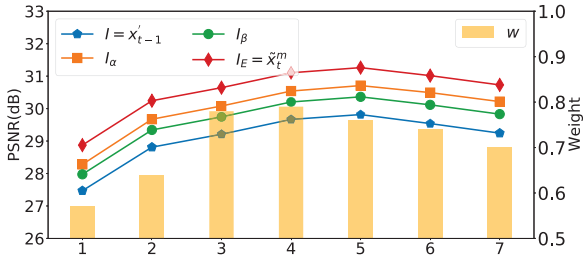

Figure 6: Comparison between the input (I), the intermediate results (I_α, I_β) and the output (I_E) of DPEG in MC. Each path has its own contributions.

effectiveness of our design, especially for the DPEG-based MC/PQE and the joint training.

Contributions of individual paths in DPEG. The DPEG network consists of an α-path and a β-path. To observe the contributions of each path, we compare the input ($I = x'_{t-1}$), the intermediate results (I_α, I_β) and the output ($I_E = \tilde{x}^m_t$) of DPEG in MC. Figure 6 presents the results obtained by averaging the 1~7-th frames of all Class D sequences. It can be seen that both I_α and I_β improve the PSNR of I. By fusing the results of I_α and I_β with the weight w ($w > 0.5$ when I_α has a better visual quality), the resulted I_E achieve a further high performance, which implies the complementarity between the two paths as well as the effectiveness of our weight fusion. By the way, the ConvLSTM also contributes to the RD performance, by reducing BDBR (in terms of PSNR) of 7.01% in Class D.

Contributions of cross-path sematic feature embedding. In Figure 3, we design the DPEG network with the sematic features $F_{t,1} \in R^{H\times W\times C}, F_{t,2} \in R^{\frac{H}{2}\times\frac{W}{2}\times C}, F_{t,3} \in R^{\frac{H}{4}\times\frac{W}{4}\times 2C}$ that are fed from α-path to β-path. To investigate the impact of this cross-path sematic feature embedding, we compare our JCEVC encoder with its reduced version without these cross-path sematic features and summarize the RD performances in Figure 7, where the results are obtained by averaging the RD performances of all Class D sequences. Compared with x265 LDP very fast, the BDBR values of

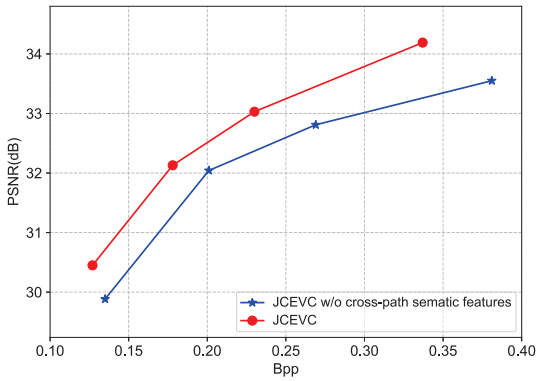

Figure 7: Comparison between JCEVC implementations with and without cross-path sematic feature embedding. The use of cross-path sematic features benefits the RD efficiency.

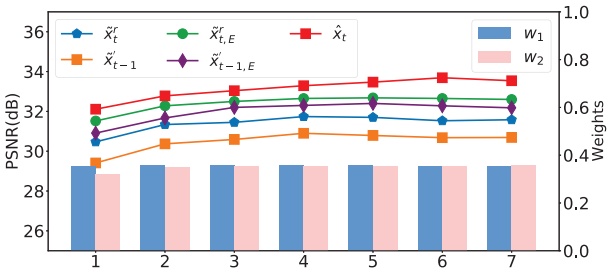

Figure 8: Comparison between the inputs ($\tilde{x}_t^r, \tilde{x}_{t-1}^{'}$), the intermediate results ($\tilde{x}_{t,E}^r, \tilde{x}_{t-1,E}^{'}$) and the output ($\hat{x}_t$) in PQE. Both inputs are enhanced and further fused to generate the reconstructed frame $\hat{x}_t$ that is of the highest quality among all these results.

the two settings are -34.26% and -44.72%, respectively. It can be easily concluded that the cross-path features $F_{t,1}, F_{t,2}, F_{t,3}$ contributes to the final coding performance.

Contributions of DPEG networks in PQE. The PQE module utilizes two DPEG networks to enhance $\tilde{x}_t^r$ and $\tilde{x}_{t-1}^{'}$ as $\tilde{x}_{t,E}^r$ and $\tilde{x}_{t-1,E}^{'}$, respectively. Then, it applies a weighted sum of $\tilde{x}_t^r$, $\tilde{x}_{t,E}^r$ and $\tilde{x}_{t-1,E}^{'}$ to obtain the final reconstruction $\hat{x}_t$. Figure 8 shows the results of these pictures by averaging the 1~7-th frames of all Class D sequences. Obviously, each DPEG network contributes to the final performance. With a weighted fusion, the finally reconstructed frame $\hat{x}_t$ is of a high visual quality in terms of PSNR. Therefore, it is reasonable to apply two DPEG networks in PQE module.

Effectiveness of loss functions. The JCEVC adopts GAN loss and MSE loss in MC and PQE modules, respectively. With different loss functions, *e.g.*, MSE loss for MC, or GAN loss for PQE, the JCEVC achieves inferior RD performances, which can be seen in Figure 9. As discussed in Section 3.4, the joint training allows a higher D in MC and further minimize it in PQE module. These different distortion constraints might be located in the attainable and unattainable regions of the perception-distortion tradeoff [5],

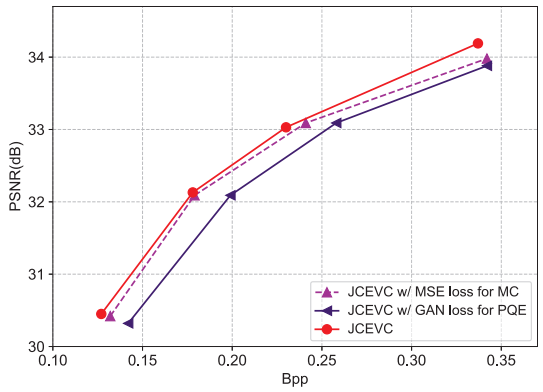

Figure 9: Comparison between JCEVC implementations with different loss functions. The RD performances of JCEVC is decreased with MSE loss for MC or GAN loss for PQE.

respectively. This can be taken as a plausible explanation that we should use different loss functions in different modules.

4.4 Limitations

The H.266/VVC incorporates enhanced block partitioning, diversified intra and inter predictions, refined ME and MC, extended transform and quantization, improved entropy coding and adaptive deblocking filters with RD optimization [7]. As compared with H.266 low delay P results in CTC [36], our JCEVC achieves a BDBR (in PSNR) of 19.31% and a BDBR (in MS-SSIM) of -29.85%. In such case, JCEVC is comparable with H.266 in terms of RD performance. However, it is still imperative to realize or imitate more advanced video coding techniques to surpass H.266. A hybrid framework to take advantages of all deep codecs is feasible. In addition, the deep video codecs prevail in the utilization of big video data. With a joint training of all modules on large-scale datasets, the deep video coding has a brighter outlook in a foreseeable future.

5 CONCLUSIONS

Nowadays, the deep video codecs have been extensively studied with ever-increasing RD performance. To compete with reigning codecs, a high-efficiency deep codec is strongly desired. In this paper, we proposed an end-to-end deep video codec called JCEVC that consists of ME, MVP, MVDC, MC, RC and PQE modules. We designed a DEPG network with dual-path generators and cross-path sematic feature embedding, and further reused it in both MC and PQE modules. Aiming at a global optimization of the RD performance, we also employed a joint training of deep video compression and enhancement. Comprehensive studies on four popular datasets have demonstrated the RD efficiency of our JCEVC method, which outperforms the state-of-the-art deep video codecs.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation of China (Grant No. 62171134) and in part by the Natural Science Foundation of Fujian Province, China (Grant No. 2022J02015).

REFERENCES

[1] 2001. Video trace library. http://trace.eas.asu.edu/yuv/index.html. (2001).

[2] 2018. BPG Image Format. https://bellard.org/bpg/. (2018).

[3] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Ballé, Sung Jin Hwang, and George Toderici. 2020. Scale-Space Flow for End-to-End Optimized Video Compression. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8500–8509. https://doi.org/10.1109/CVPR42600.2020.00853

[4] Gisle Bjontegaard. 2001. Calculation of average PSNR differences between RD-Curves. *Doc. VCEG-M33, ITU-T Video Coding Experts Group (VCEG)* (Jan. 2001).

[5] Yochai Blau and Tomer Michaeli. 2018. The Perception-Distortion Tradeoff. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6228–6237. https://doi.org/10.1109/CVPR.2018.00652

[6] Frank Bossen. 2013. Common test conditions and software reference configurations. *Doc. JCTVC-L1100, Joint Collaborative Team on Video Coding (JCT-VC)* (Jan. 2013).

[7] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. 2021. Overview of the Versatile Video Coding (VVC) Standard and its Applications. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 10 (2021), 3736–3764. https://doi.org/10.1109/TCSVT.2021.3101953

[8] Chunlei Cai, Li Chen, Xiaoyun Zhang, and Zhiyong Gao. 2020. End-to-End Optimized ROI Image Compression. *IEEE Transactions on Image Processing* 29 (2020), 3442–3457. https://doi.org/10.1109/TIP.2019.2960869

[9] Jianrui Cai, Zisheng Cao, and Lei Zhang. 2020. Learning a Single Tucker Decomposition Network for Lossy Image Compression With Multiple Bits-per-Pixel Rates. *IEEE Transactions on Image Processing* 29 (2020), 3612–3625. https://doi.org/10.1109/TIP.2020.2963956

[10] Li-Heng Chen, Christos G. Bampis, Zhi Li, Andrey Norkin, and Alan C. Bovik. 2021. ProxIQA: A Proxy Approach to Perceptual Optimization of Learned Image Compression. *IEEE Transactions on Image Processing* 30 (2021), 360–373. https://doi.org/10.1109/TIP.2020.3036752

[11] Tong Chen, Haojie Liu, Zhan Ma, Qiu Shen, Xun Cao, and Yao Wang. 2021. End-to-End Learnt Image Compression via Non-Local Attention Optimization and Improved Context Modeling. *IEEE Transactions on Image Processing* 30 (2021), 3179–3191. https://doi.org/10.1109/TIP.2021.3058615

[12] Yu-Sheng Chen, Yu-Ching Wang, Man-Hsin Kao, and Yung-Yu Chuang. 2018. Deep Photo Enhancer: Unpaired Learning for Image Enhancement from Photographs with GANs. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*. 6306–6314. https://doi.org/10.1109/CVPR.2018.00660

[13] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. 2020. Learned Image Compression With Discretized Gaussian Mixture Likelihoods and Attention Modules. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 7936–7945. https://doi.org/10.1109/CVPR42600.2020.00796

[14] Zhenyu Guan, Qunliang Xing, Mai Xu, Ren Yang, Tie Liu, and Zulin Wang. 2021. MFQE 2.0: A New Approach for Multi-Frame Quality Enhancement on Compressed Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 3 (2021), 949–963. https://doi.org/10.1109/TPAMI.2019.2944806

[15] Amirhossein Habibian, Ties Van Rozendaal, Jakub Tomczak, and Taco Cohen. 2019. Video Compression With Rate-Distortion Autoencoders. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 7032–7041. https://doi.org/10.1109/ICCV.2019.00713

[16] Muhammad Haris, Greg Shakhnarovich, and Norimichi Ukita. 2020. Space-Time-Aware Multi-Resolution Video Enhancement. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2856–2865. https://doi.org/10.1109/CVPR42600.2020.00293

[17] Dailan He, Yaoyan Zheng, Baocheng Sun, Yan Wang, and Hongwei Qin. 2021. Checkerboard Context Model for Efficient Learned Image Compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 14766–14775. https://doi.org/10.1109/CVPR46437.2021.01453

[18] Zhihao Hu, Zhenghao Chen, Dong Xu, Guo Lu, Wanli Ouyang, and Shuhang Gu. 2020. Improving Deep Video Compression by Resolution-Adaptive Flow Coding. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 193–209. https://doi.org/10.1007/978-3-030-58536-5_12

[19] Zhihao Hu, Guo Lu, and Dong Xu. 2021. FVC: A New Framework towards Deep Video Compression in Feature Space. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1502–1511. https://doi.org/10.1109/CVPR46437.2021.00155

[20] Nick Johnston, Damien Vincent, David Minnen, Saurabh Covell, Michele Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. 2018. Improved Lossy Image Compression with Priming and Spatially Adaptive Bit Rates for Recurrent Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4385–4393. https://doi.org/10.1109/CVPR.2018.00461

[21] Jan P. Klopp, Keng-Chi Liu, Shao-Yi Chien, and Liang-Gee Chen. 2021. Online-Trained Upsampler for Deep Low Complexity Video Compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 7929–7938. https://doi.org/10.1109/ICCV48922.2021.00783

[22] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. 2019. Context-adaptive Entropy Model for End-to-end Optimized Image Compression. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[23] Chongyi Li, Chunle Guo, and Change Loy Chen. 2021. Learning to Enhance Low-Light Image via Zero-Reference Deep Curve Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1. https://doi.org/10.1109/TPAMI.2021.3063604

[24] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. 2020. M-LVC: Multiple Frames Prediction for Learned Video Compression. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 3543–3551. https://doi.org/10.1109/CVPR42600.2020.00360

[25] Bowen Liu, Yu Chen, Shiyu Liu, and Hun-Seok Kim. 2021. Deep Learning in Latent Space for Video Prediction and Compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 701–710. https://doi.org/10.1109/CVPR46437.2021.00076

[26] Guo Lu, Chunlei Cai, Xiaoyun Zhang, Li Chen, Wanli Ouyang, Dong Xu, and Zhiyong Gao. 2020. Content Adaptive and Error Propagation Aware Deep Video Compression. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 456–472. https://doi.org/10.1007/978-3-030-58536-5_27

[27] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. 2019. DVC: An End-To-End Deep Video Compression Framework. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 10998–11007. https://doi.org/10.1109/CVPR.2019.01126

[28] Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Li Chen, Zhiyong Gao, and Dong Xu. 2021. An End-to-End Learning Framework for Video Compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 10 (2021), 3292–3308. https://doi.org/10.1109/TPAMI.2020.2988453

[29] Haichuan Ma, Dong Liu, Ning Yan, Houqiang Li, and Feng Wu. 2020. End-to-End Optimized Versatile Image Compression With Wavelet-Like Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1–1. https://doi.org/10.1109/TPAMI.2020.3026003

[30] Alexandre Mercat, Marko Viitanen, and Jarno Vanne. 2020. UVG dataset: 50/120fps 4K sequences for video codec analysis and development. In *MMSys '20: 11th ACM Multimedia Systems Conference*. 297–302. https://doi.org/10.1145/3339825.3394937

[31] Zhangkai Ni, Wenhan Yang, Shiqi Wang, Lin Ma, and Sam Kwong. 2020. Towards Unsupervised Deep Image Enhancement With Generative Adversarial Network. *IEEE Transactions on Image Processing* 29 (2020), 9140–9151. https://doi.org/10.1109/TIP.2020.3023615

[32] Anurag Ranjan and Michael J. Black. 2017. Optical Flow Estimation Using a Spatial Pyramid Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2720–2729. https://doi.org/10.1109/CVPR.2017.291

[33] Oren Rippel, Alexander G. Anderson, Kedar Tatwawadi, Sanjay Nair, Craig Lytle, and Lubomir Bourdev. 2021. ELF-VC: Efficient Learned Flexible-Rate Video Coding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 14479–14488. https://doi.org/10.1109/ICCV48922.2021.01421

[34] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander Anderson, and Lubomir Bourdev. 2019. Learned Video Compression. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 3453–3462. https://doi.org/10.1109/ICCV.2019.00355

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *2015 Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

[36] Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu. 2021. Temporal Context Mining for Learned Video Compression. *arXiv e-prints* (2021).

[37] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-Chun Woo. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. MIT Press, 802–810. https://doi.org/10.1007/978-3-319-21233-3_6

[38] Taeyoung Son, Juwon Kang, Namyup Kim, Sunghyun Cho, and Suha Kwak. 2020. URIE: Universal Image Enhancement for Visual Recognition in the Wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 749–765. https://doi.org/10.1007/978-3-030-58545-7_43

[39] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. 2012. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (2012), 1649–1668. https://doi.org/10.1109/TCSVT.2012.2221191

[40] David.S. Taubman and Michael.W. Marcellin. 2002. JPEG2000: standard for interactive imaging. *Proc. IEEE* 90, 8 (2002), 1336–1357. https://doi.org/10.1109/JPROC.2002.800725

[41] G.K. Wallace. 1992. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics* 38, 1 (1992), xviii–xxxiv. https://doi.org/10.1109/30.125072

[42] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C.-C. Jay Kuo. 2016. MCL-JCV: A JND-based H.264/AVC video quality assessment dataset. In *2016 IEEE International Conference on Image Processing (ICIP)*. 1509–1513. https://doi.org/10.1109/ICIP.2016.7532610

[43] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman. 2019. Video Enhancement with Task-Oriented Flow. *International Journal of Computer Vision* 127, 8 (2019), 1106–1125. https://doi.org/10.1007/s11263-018-01144-2

[44] Fei Yang, Luis Herranz, Yongmei Cheng, and Mikhail G. Mozerov. 2021. Slimmable Compressive Autoencoders for Practical Neural Image Compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 4996–5005. https://doi.org/10.1109/CVPR46437.2021.00496

[45] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. 2020. Learning for Video Compression With Hierarchical Quality and Recurrent Enhancement. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6627–6636. https://doi.org/10.1109/CVPR42600.2020.00666

[46] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. 2021. Learning for Video Compression With Recurrent Auto-Encoder and Recurrent Probability Model. *IEEE Journal of Selected Topics in Signal Processing* 15, 2 (2021), 388–401. https://doi.org/10.1109/JSTSP.2020.3043590

[47] Ren Yang, Mai Xu, Zulin Wang, and Tianyi Li. 2018. Multi-frame Quality Enhancement for Compressed Video. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*. 6664–6673. https://doi.org/10.1109/CVPR.2018.00697

[48] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. 2018. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 286–301. https://doi.org/10.1007/978-3-030-01234-2_18