Conditional Coding and Variable Bitrate for Practical Learned Video Coding

Théo Ladune, Pierrick Philippe
Orange
Rennes, France
firstname.lastname@orange.com

Wassim Hamidouche, Lu Zhang, Olivier Déforges
CNRS, IETR – UMR 6164
Univ. Rennes, INSA Rennes
firstname.lastname@insa-rennes.fr

Abstract

This paper introduces a practical learned video codec. Conditional coding and quantization gain vectors are used to provide flexibility to a single encoder/decoder pair, which is able to compress video sequences at a variable bitrate. The flexibility is leveraged at test time by choosing the rate and GOP structure to optimize a rate-distortion cost. Using the CLIC21 video test conditions, the proposed approach shows performance on par with HEVC.

1. Introduction

Deep neural networks allow to perform complex non-linear transforms. In image coding, these transforms are learned end-to-end [2] to minimize a rate-distortion cost. This leads to compression performance competitive with the image coding configuration of VVC [6], the latest ITU/MPEG video coding standard. Video coding is a more challenging task than image coding due to the additional temporal redundancies, often removed through motion compensation. It consists in computing a temporal prediction from already decoded frames to only send the unpredictable part. Frames coded using already decoded ones are called *inter* frames, others are called *intra* frames.

The unpredictable content is often conveyed through residual coding *i.e.* by sending the difference between the frame and its temporal prediction. In the literature, this results in a *two-codec* approach, with an inter-frame codec dedicated to residual signal and an intra-frame codec for image-domain signal [1, 12, 13]. Moreover, previous learned video coders [1, 8, 12] are designed to operate under a single rate constraint (*i.e.* a single quality level). As such, having several quality levels available requires the storage of one decoder per quality level, which is not practical for real world application.

This article introduces an end-to-end learned factorized system. It is claimed to be a practical video coder as it provides some essential real-world features. First, a single coder processes intra and inter frames, allowing to use any GOP structures (intra/inter arrangement). Second, a single coder allows to continuously select the rate of each frame, enabling the accurate optimization of each frame RD cost.

The coding scheme is based on two networks, MOFNet and CodecNet [7]. MOFNet conveys side-information (prediction parameters, coding mode) while CodecNet transmits the unpredictable content. Both networks use conditional coding to exploit the decoder-side information while being resilient to its absence. This enables the processing of inter and intra frames with the same coder. Gain vectors [5] are added at the quantization step of both networks to adapt to different rate constraints. The system is shown to be competitive with the video coding standard HEVC [9] under the CLIC21 video test conditions [11].

2. Coding Scheme

Let $\{\mathbf{x}_i \in \mathbb{R}^{C\times H\times W}, i \in \mathbb{N}\}$ be a video with C color channels of height H and width W[1]. In this work, videos are encoded using one intra frame (I-frame) followed by several Groups of Pictures (GOP). A GOP is a fixed pattern of inter-frames, made of P-frames (use one already decoded frame as reference) and B-frames (use two references). The Fig. 1 shows an I-frame and two different GOP structures.

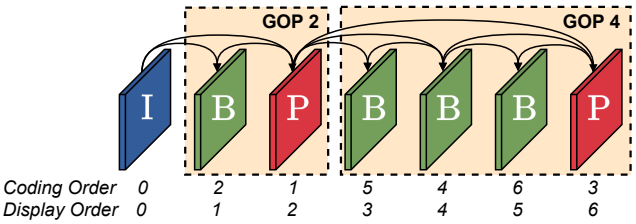


Figure 1: Two different GOP structures, GOP2 and GOP4

[1] Videos are in YUV 420. For convenience, a bilinear upsampling is used to obtain YUV 444 data.

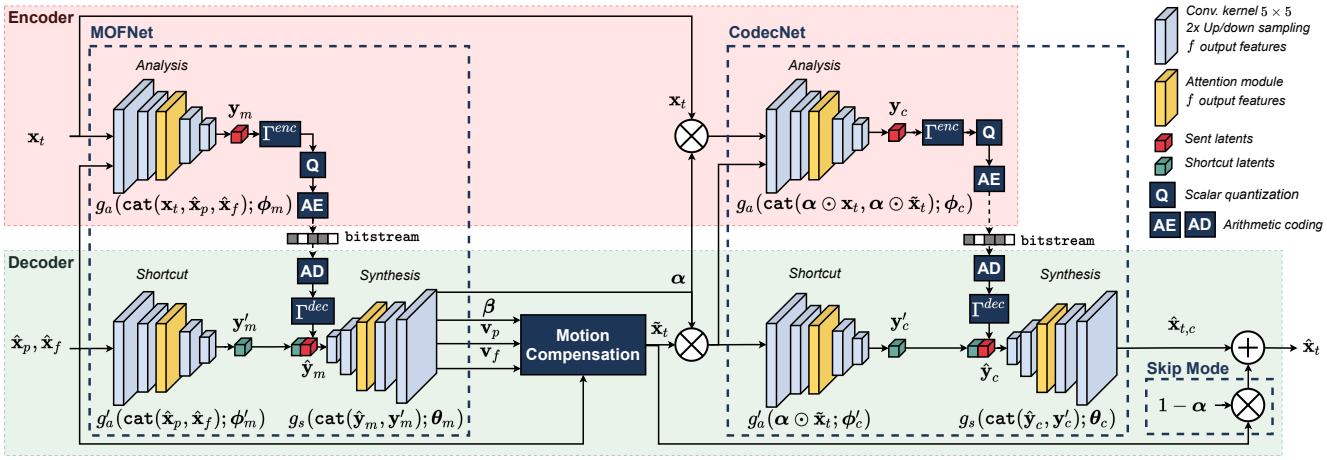


Figure 2: Diagram of the system. Latents probability model uses hyperpriors [3] and is omitted for clarity. Attention modules are implemented as proposed in [4] and $f = 192$. There are 28 millions learnable parameters $\{\boldsymbol{\phi}, \boldsymbol{\theta}\}$.

Video sequences are compressed to minimize a rate-distortion trade-off weighted by λ:

$$\mathcal{L}_\lambda = \sum_t \mathrm{D}(\hat{\mathbf{x}}_t, \mathbf{x}_t) + \lambda \mathrm{R}(\hat{\mathbf{x}}_t), \tag{1}$$

with D a distortion measure between the original frame $\mathbf{x}_t$ and the coded one $\hat{\mathbf{x}}_t$ and $\mathrm{R}(\hat{\mathbf{x}}_t)$ the rate associated to $\hat{\mathbf{x}}_t$. Following the CLIC21 video test conditions, D is based on the MS-SSIM [10]: $\mathrm{D}(\hat{\mathbf{x}}_t, \mathbf{x}_t) = 1 - \text{MS-SSIM}(\hat{\mathbf{x}}_t, \mathbf{x}_t)$.

2.1. Overview

Let $\mathbf{x}_t$ be a frame to code. To reduce its rate, the system leverages up to two reference frames denoted as $(\hat{\mathbf{x}}_p, \hat{\mathbf{x}}_f)$. The proposed coding scheme, shown in Fig. 2, is split between two convolutional networks, MOFNet and CodecNet. MOFNet takes $(\mathbf{x}_t, \hat{\mathbf{x}}_p, \hat{\mathbf{x}}_f)$ as inputs to compute and convey two optical flows $(\mathbf{v}_p, \mathbf{v}_f)$, a pixel-wise prediction weighting $\boldsymbol{\beta}$ and a pixel-wise coding mode selection α. Each optical flow represents a pixel-wise motion from $\mathbf{x}_t$ to one of the reference, used to interpolate the reference using a bilinear warping. Both reference warpings are summed and weighted by $\boldsymbol{\beta}$ to obtain a temporal prediction $\tilde{\mathbf{x}}_t$:

$$\tilde{\mathbf{x}}_t = \boldsymbol{\beta} \odot w(\hat{\mathbf{x}}_p; \mathbf{v}_p) + (1 - \boldsymbol{\beta}) \odot w(\hat{\mathbf{x}}_f; \mathbf{v}_f), \tag{2}$$

where w is a bilinear warping, $\odot$ a pixel-wise multiplication, $\boldsymbol{\beta} \in [0,1]^{H \times W}$ and $\mathbf{v}_p, \mathbf{v}_f \in \mathbb{R}^{2 \times H \times W}$.

The coding mode selection $\boldsymbol{\alpha} \in [0,1]^{H \times W}$ arbitrates between two coding modes: *Skip* mode (direct copy of $\tilde{\mathbf{x}}_t$) or CodecNet to transmit $\mathbf{x}_t$ using information from $\tilde{\mathbf{x}}_t$ to reduce the rate. The contributions from the two coding modes are summed to obtain the reconstructed frame $\hat{\mathbf{x}}_t$:

$$\hat{\mathbf{x}}_t = \underbrace{(1 - \boldsymbol{\alpha}) \odot \tilde{\mathbf{x}}_t}_{\text{Skip}} + \underbrace{c(\boldsymbol{\alpha} \odot \mathbf{x}_t, \boldsymbol{\alpha} \odot \tilde{\mathbf{x}}_t)}_{\text{CodecNet}}. \tag{3}$$

The total rate is the sum of MOFNet and CodecNet rate. When there is no reference available, MOFNet is bypassed, $\tilde{\mathbf{x}}_t$ is set to 0 and $\boldsymbol{\alpha}$ to 1: image coding is de facto used.

2.2. Conditional Coding

MOFNet and CodecNet rely on conditional coding [7] to better exploit decoder-side information than residual coding. Conditional coding supplements the analysis and synthesis transforms of the auto-encoder architecture [3] with a *shortcut* transform. The shortcut transform retrieves information from the references (*i.e.* at no rate) as latents $\mathbf{y}'$. Thus, the analysis transform sends only the information not present at the decoder as latents $\mathbf{y}$. Finally, the synthesis transform concatenates both latents as inputs to compute its output. After training, conditional coding is flexible enough to work with zero, one or two references. This makes the proposed coder able to process all types of frames, allowing to use any GOP structure composed of I, P and B-frames.

2.3. Variable rate

The system operates at variable rates using scaling operations in the quantization stage [5]. The coder is trained simultaneously for N rate constraints $\{\lambda_1, \ldots, \lambda_N\}$. Each λ_i is associated to a pair of learned feature-wise gain vectors $\Gamma_i^{enc}, \Gamma_i^{dec} \in \mathbb{R}^{C_y}$, with C_y the number of channels of the latents $\hat{\mathbf{y}}$. The gain vectors are applied through feature-wise multiplication (see Fig. 4).

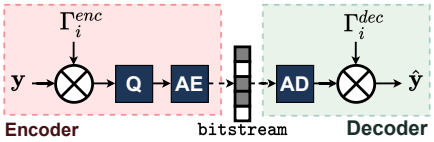
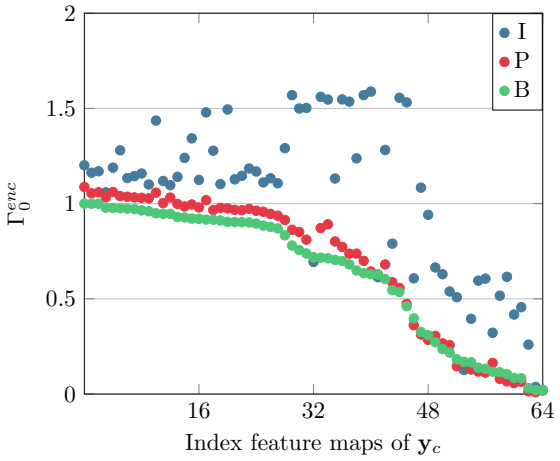
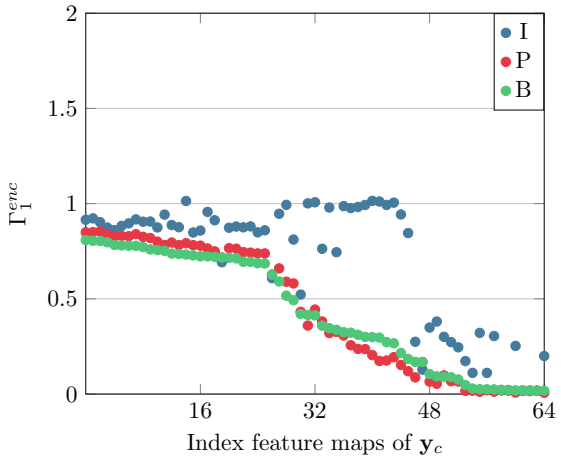


Figure 4: Usage of the quantization gain vectors.



(a) Γ_0^{enc} learned under the rate constraint λ_0 (higher rate).



(b) Γ_1^{enc} learned under the rate constraint λ_1 (lower rate).

Figure 3: Feature-wise value of the CodecNet gain vectors Γ_0^{enc} and Γ_1^{enc}, corresponding to two rate constraints λ_0 (higher rate) and λ_1 (lower rate). The graphs present the gain vectors associated to I, P and B-frames.

The gain vectors can be thought of as feature-wise adapted quantization steps. Since the quantization steps of I, P and B frames may be different, a dedicated gain vector pair is learned for each frame type and for each rate constraint, for both MOFNet and CodecNet. After training, it is possible to operate under any *continuous* rate constraint $r \in [1, N]$ by interpolating the gain vectors through a weighted geometric averaging:

$$\Gamma_r = \Gamma_{\lfloor r \rfloor}^{1-l} \odot \Gamma_{\lceil r \rceil}^{l}, \text{ with } l = r - \lfloor r \rfloor. \tag{4}$$

3. Training

The purpose of the training phase is to prepare the system to code I, P and B-frames under $N = 6$ rate constraints. To this effect, it is trained on the smallest coding structure featuring the 3 types of frames *i.e.* an I-frame followed by a GOP2 (see Fig. 1). For each training iteration, a rate index $i \in \{1, N\}$ is randomly selected. Then the 3 frames are coded using the corresponding Γ_i, followed by a single back-propagation to minimize the rate-distortion cost of eq. (1) using λ_i. The N rate constraints are chosen to be distributed around the 1 Mbit/s rate target of CLIC21.

The learning process is performed through a rate-distortion loss. No element of the system requires a pre-training or a dedicated loss. Moreover, coding the 3 frames in the forward pass makes the system able to deal with coded references, leading to better coding performance.

4. Experimental Results

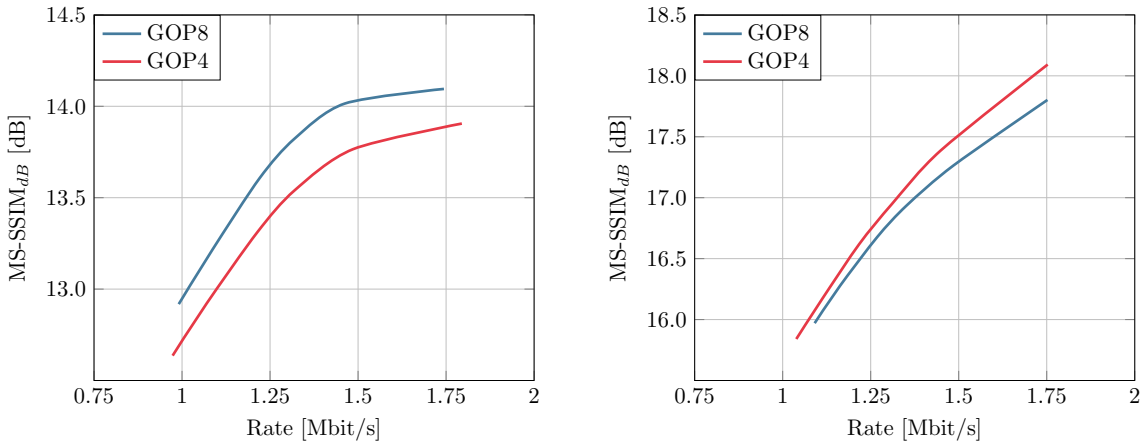
4.1. Gain Vector Visualization

The encoder gain vectors of CodecNet obtained after training are illustrated in Figures 3a (higher rate) and 3b (lower rate). Γ^{enc} for the I-frames are always larger than those of the P-frames which are almost identical to those of the B-frames. As a bigger Γ^{enc} translates to a more accurate quantization of $\mathbf{y}$, it means that I-frames require more precision (*i.e.* more rate) than the P or B-frames. This is because I-frames are exclusively conveyed by CodecNet latents $\mathbf{y}_c$ whereas P and B-frames rely on Skip Mode and CodecNet shortcut transform, allowing them to use a less accurate quantization step. Comparing Γ_0^{enc} and Γ_1^{enc} illustrates that rate control is achieved through changes in the quantization precision. As expected, a less accurate quantization is used at lower rate.

4.2. Rate-distortion Results

The rate and GOP structure flexibility provided by the coder allows to tune these two parameters to optimize the rate-distortion (RD) cost individually for each video. The Fig. 5 shows the RD curves of the coder on two sequences from the CLIC21 validations set for GOP4 and GOP8 structures. This example shows that there is no consistent best GOP structure. Thanks to its flexibility, the proposed coder can test different GOP structures during the encoding process and select the best for each sequence, resulting in better overall performances. Gain vectors interpolation presented in eq. (4) makes the coder able to target any rate, allowing to obtain continuous RD curves.

The proposed system is evaluated under the CLIC21 video test conditions. The objective is to get the highest MS-SSIM at about 1 Mbit/s. The CLIC21 validation set contains 100 videos with 60 frames. The coder flexibility is leveraged by optimizing the rate and the GOP structure of each sequence according to a global RD cost. The



(a) *VR_2160P-5489*. Further reference frames are beneficial. (b) *VR_1080P-384e*. Further reference frames are harmful.

Figure 5: Rate-distortion curves on two CLIC21 sequences for two different GOP structures. The intra period is set to 32 frames for both GOP structures. The quality is measured as MS-SSIM$_{dB} = -10\log_{10}(1-\text{MS-SSIM})$, higher is better.

Hyperprior curve in Fig. 6 shows the performance of the coder against two implementations of the video coding standard HEVC: the widely used `x265`[2] and the HM 16.22, the reference HEVC coder. The proposed system offers compelling compression results, consistently outperforming `x265` from 0.75 to 1.25 Mbit/s.

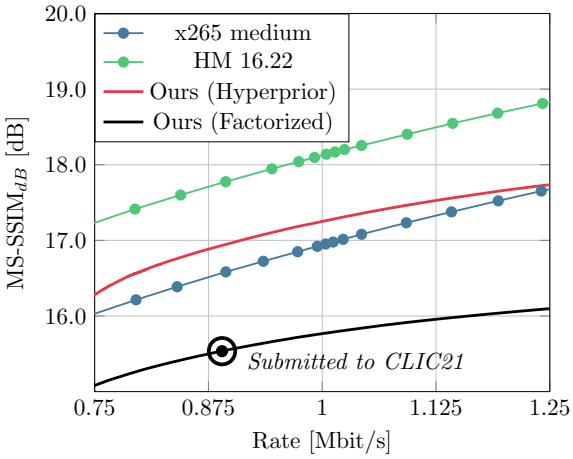


Figure 6: Rate-distortion curves on the CLIC21 video validation set. Two versions of the system are compared against two implementations of HEVC. The quality is measured as MS-SSIM$_{dB} = -10\log_{10}(1-\text{MS-SSIM})$, higher is better.

[2] `ffmpeg -video_size WxH -i in.yuv -c:v libx265 -pix_fmt yuv420p -crf QP -preset medium -tune ssim out.mp4`

5. CLIC21 Video Track

A system alike to the one described in this article has been submitted to the CLIC21 video track under the name E2E_T_OL. Because of the model size penalty, the number of convolution features f is reduced from 192 to 128. In order to ensure cross-platform arithmetic coding, the submitted coder rely on a simpler latents probability model, replacing the hyperprior mechanism with a factorized model [3]. The degradation of compression efficiency associated to the factorized model is shown in Fig 6.

Table 1: CLIC21 leaderboard results validation set

System name	MS-SSIM	Size [KBytes] Data	Size [KBytes] Model	Decoding time [s]
E2E_T_OL	0.97204	23769	45235	13356

6. Conclusion

This paper introduces a end-to-end learned video coder, integrating many features required for a practical usage. Thanks to conditional coding and quantization gain vectors, a single encoder/decoder pair is able to process all types of frame (I, P & B) at any continuous rate target. Aside from being more convenient, this flexibility brings performance gains through rate and GOP structure competition. The relevance of the proposed approach is proved by outperforming `x265` on the CLIC21 video coding track.

Immediate future work will consist in the quantization of the network operations in order allowing the usage of hyperpriors in a cross-platform situation.

References

[1] Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[2] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France*, 2017.

[3] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada*, 2018.

[4] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules, 2020.

[5] T. Guo, J. Wang, Z. Cui, Y. Feng, Y. Ge, and B. Bai. Variable rate image compression with content adaptive optimization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 533–537, 2020.

[6] S. Kim J. Chen, Y. Ye. Algorithm description for versatile video coding and test model 8 (vtm 8), Jan. 2020.

[7] Théo Ladune, Pierrick Philippe, Wassim Hamidouche, Lu Zhang, and Olivier Déforges. Conditional coding for flexible learned video compression. In *Neural Compression: From Information Theory to Applications – Workshop @ ICLR 2021*, 2021.

[8] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. DVC: an end-to-end deep video compression framework. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, 2019*, pages 11006–11015, 2019.

[9] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Trans. Cir. and Sys. for Video Technol.*, 22(12):1649–1668, Dec. 2012.

[10] Zhou Wang, Eero P. Simoncelli, and Alan C. Bovik. Multiscale structural similarity for image quality assessment. In *in Proc. IEEE Conf. on Signals, Systems, and Computers*, pages 1398–1402, 2003.

[11] Workshop and Challenge on Learned Image Compression. https://www.compression.cc/, June 2021.

[12] Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with hierarchical quality and recurrent enhancement, 2020.

[13] M. Akin Yilmaz and A. Murat Tekalp. End-to-end rate-distortion optimization for bi-directional learned video compression, 2020.