GOP-BASED LATENT REFINEMENT FOR LEARNED VIDEO CODING

Mohsen Abdoli, Gordon Clare, Félix Henry

GOP-BASED LATENT REFINEMENT FOR LEARNED VIDEO CODING

Mohsen Abdoli, Gordon Clare and Félix Henry

IRT b-com, 1219 Avenue des Champs Blancs, 35510 Cesson-Sévigné, France.

ABSTRACT

This paper presents a method allowing learned video encoders to apply arbitrary latent refinement strategies to serve as Rate-Distortion Optimization (RDO) at the time of encoding. To do so, a latent domain search is applied on an initial latent representation of the video signal. This search is implemented as a set of iterations, each of which performs a gradient descent with back-propagation of error defined by a Lagrangian RD cost. This cost function is intentionally chosen to be the same as the cost function that was used during the end-to-end model training, except that instead of updating model weights, each iteration fine-tunes the latent representation itself. Moreover, a temporal look-ahead is integrated in the cost function of I and P frames to take into account the cascade effect of their latent fine-tuning on subsequent frames in the Group of Pictures (GOP). The experiments show that the proposed latent space RDO method can improve by 11.6% and 9.4% in terms of BD-BR coding efficiency in Random-Access (RA) and All-Intra (AI) configurations, when applied on top a high-performance open-source end-to-end codec.

***Index Terms*—Learned Video Coding, Rate-Distortion Optimization, Back-propagation with gradient decent.**

I. INTRODUCTION

End-to-end Learned Video Codec (LVC) systems have recently emerged to challenge conventional coding systems that have been widely used for decades [1]–[4]. Even though different sectors of the media broadcast chain still seem unprepared to consider their deployment, these alternative coding systems are rapidly maturing and attracting attention. Some of the notable challenges to address so as to pave the way for deployment of LVC systems are: further improving compression efficiency, hardware support (particularly at the decoder-side), standardization of different aspects, and peripheral encoder functionalities which normally allow a codec to be used in real-world scenarios [5].

One of the critical peripheral functionalities of a codec is the the ability to apply arbitrary Rate-Distortion Optimization (RDO) strategies at the time of encoding. In LVC, once a codec model is trained, its utilisation is fixed unbending as two forward inference passes by the encoder $\mathbf{E}$ and decoder $\mathbf{D}$. This simplicity comes at the cost of flexibility and limited choices, compared to conventional video coders, such as High Efficiency Video Coding (HEVC) and Versatile Video Coding (VVC), where RDO algorithms are in essence extremely flexible and are used widely and arbitrarily.

A lack of effective RDO schemes is one of the aspects blocking LVC systems from being realistically assessed in real-world scenarios [6]. There have been studies addressing this shortcoming. A first approach is to train the encoder model such that it performs RDO in inference mode. For instance, RDOnet deploys masking layers to zero-out certain coefficients. By training models with such layers, unimportant regions of the image are identified during inference and do not have their information transmitted [7]–[9]. The drawback of this approach is the lack of inference-time signal adaptation. One solution to enable such a feature is to mimic conventional block-based RDO using mode-selection and choosing the mode that minimizes the Rate-Distortion Cost (RD-cost). Examples of how to define "coding modes" at the block-level are varying block resolution [10], or training different probability distribution models [11]. Online model fine-tuning and transmission of the refined model as meta-data is yet another approach for adaptively encoding a sequence [12, 13]. Despite their promising performance, model fine-tuning methods suffer from heavy encoding-time computation. Latent adaptation methods that are used in Learned Image Codec (LIC) systems, on the other hand, fine-tune the signal itself, rather than models. One realization of this concept is implemented by a pre-encoding phase in which the input image is modified in the pixel domain [14, 15]. Alternatively, one can modify the latent representation of an image after encoding and before quantization [16, 17]. In both cases, these methods apply back-propagation with gradient descent on the RD-cost of the encoded image latent representation and obtain an alternative representation with a better coding efficiency.

Inspired by the LIC latent adaptation concept, this paper enables RDO in the context of LVC. To do so, each compressed frame of video is fine-tuned in the latent domain to adapt with content. In order to take into account the Group of Pictures (GOP) structure of a video, a short look-ahead is also deployed to incorporate the impact of fine-tuning the latent representation of the current frame on future frames. The rest of this paper is organized as follows. First, a base coder is formulated in Section II on top of which the proposed method is presented. Section III elaborates the proposed method, introducing latent space fine-tuning RDO. Experimental results are presented in Section IV and finally the paper is concluded in Section V.

II. BASE CODER

Since the proposed latent domain RDO is essentially orthogonal to the generic design of its underlying LVC, it is assumed here that a pre-trained codec tuple is given in the form of an encoder and a decoder models, respectively expressed as $< \mathbf{E}, \mathbf{D} >$, whose parameters $\Theta =< \Theta_e, \Theta_d >$ are jointly optimized. Using this LVC, encoding a frame x transforms it to a latent representation z in a latent space, as:

$$z = \mathbf{E}(x; \Theta_e). \tag{1}$$

Subsequently, decoding the latent representation z transforms it back to a pixel-domain representation $\hat{x}$:

$$\hat{x} = \mathbf{D}(z; \Theta_d). \tag{2}$$

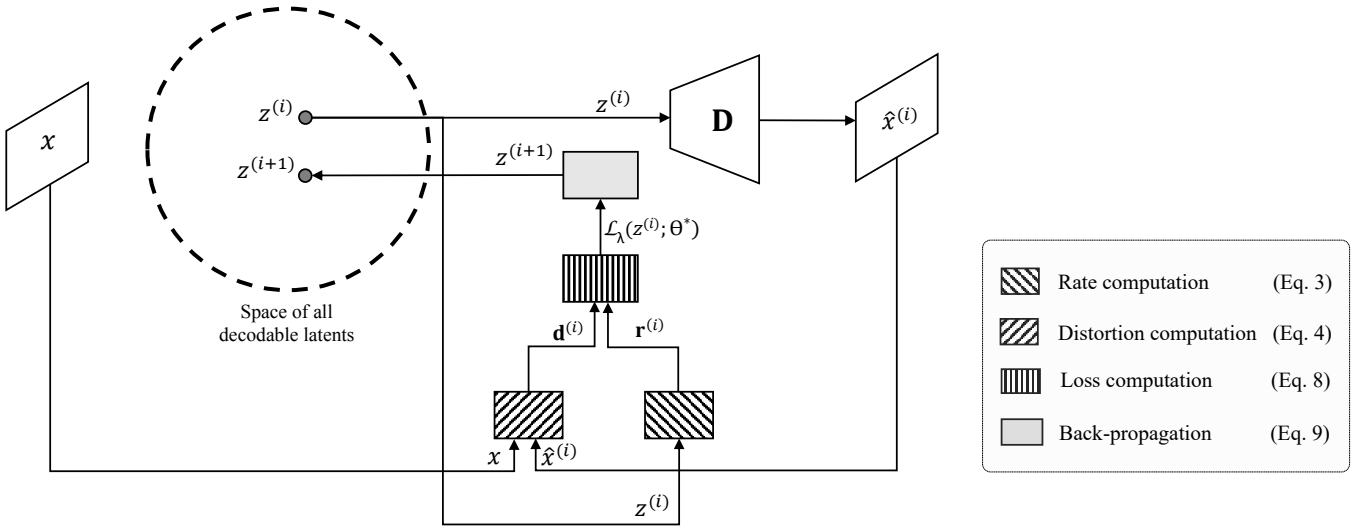

Fig. 1: One iteration of the proposed frame-based RDO method. $z^{(i)}$ and $z^{(i+1)}$ are the input and output of this process, respectively, while $z^{(i+1)}$ is also the input to the next iteration.

The above round-trip is typically monitored by the two metrics of rate and distortion. Depending on available information, the rate metric is either the exact number of bits, or an estimation of it. Given an entropy coder $\mathbf{EC}$, the exact rate of a latent representation can be computed as $\mathbf{r}(z) = \mathbf{EC}(z)$. However, the entropy coder operates exclusively on discrete data, thus is non-differentiable. Consequently, during the training process of an LVC, the above rate is typically estimated by an approximated Probability Distribution Function (PDF) of the continuous uniform distribution p, as:

$$\tilde{\mathbf{r}}(z) = -\log_2 p_z(z). \tag{3}$$

Moreover, the distortion metric $\mathbf{d}$ is computed as the squared l^2-norm of the compression loss error:

$$\mathbf{d}(x, z) = \|x - \hat{x}\|_2^2 = \|x - \mathbf{D}(z; \Theta_d)\|_2^2. \tag{4}$$

Given a Lagrangian multiplier λ, the training of the codec is performed on a dataset of samples $\mathbf{x} = \{x_k | k = 0, 1, ..., |\mathbf{x}| - 1\}$, by minimizing an estimation of its rate-distortion cost as the loss function:

$$\mathcal{L}_\lambda(\mathbf{x}; \Theta) = \frac{1}{|\mathbf{x}|} \sum_{x_k \in \mathbf{x}} \mathbf{d}(x_k, z_k) + \lambda \tilde{\mathbf{r}}(z_k). \tag{5}$$

The optimization of the model parameters Θ to minimize the loss function of Eq. 5 is typically carried out by iteratively applying gradient descent with back-propagation of its error, expressed as $\nabla_\Theta \mathcal{L}(\Theta)$ in Eq. 6. Each iteration j of this algorithm back-propagates the error with a given learning rate parameter, denoted as η:

$$\Theta^{(j+1)} = \Theta^{(j)} - \eta \nabla_\Theta \mathcal{L}(\mathbf{x}; \Theta^{(j)}). \tag{6}$$

III. PROPOSED LATENT SPACE RDO

Frame-based RDO with latent fine-tuning

It is assumed that an operational codec tuple is provided as $< \mathbf{E}, \mathbf{D} >$, whose optimal parameters Θ^* are optimized by using Eq. 6 on a dataset. The proposed method is applied in an iterative manner on individual frames of a sequence. Each iteration i, starts with an initial latent representation $z^{(i)}$ and ends with an output latent representation $z^{(i+1)}$, which will serve as the initial latent representation of the next iteration. In this section, this repetitive process is described for the first iteration (*i.e.* $i = 0$), where the initial latent representation is exceptionally generated by the forward pass of the encoder:

$$z^{(0)} = \mathbf{E}(x; \Theta_e^*) \tag{7}$$

As the rest of the process is common for all remaining iterations, the initial latent representation of the ongoing iteration is denoted as $z^{(i)}$. By computing the approximate rate (*i.e.* Eq. 3) and the distortion (*i.e.* Eq. 4) of a single latent sample $z^{(i)}$, one can obtain its rate-distortion cost based on the given Lagrangian multiplier λ:

$$\mathcal{L}_\lambda(z^{(i)}; \Theta^*) = \mathbf{d}(x, z^{(i)}) + \lambda \tilde{\mathbf{r}}(z^{(i)}). \tag{8}$$

The gradient of the loss function with respect to the latent representation z is expressed as $\nabla_z \mathcal{L}_\lambda(\Theta)$. In contrast to $\nabla_\Theta \mathcal{L}_\lambda(\Theta)$ that is computed with respect to model parameters Θ during the training phase, the gradient used in the RDO is computed with respect to the latents of the input signal. As a result, back-propagation of this error updates only the latents and keeps Θ unchanged. This process is expressed as follows:

$$z^{(i+1)} = z^{(i)} - \eta \nabla_z \mathcal{L}_\lambda(z^{(i)}; \Theta^*) \tag{9}$$

By choosing the same value of λ as in the training phase, the above computation turns into an RDO process that is aligned with the end-to-end training of the codec models. Since this optimization is aligned with that of the training phase, each iteration actually fine-tunes the latent representation $z^{(i+1)}$.

Fig. 1 schematically summarizes the above process for one iteration. It is noteworthy that, except in producing the initial latent representation $z^{(0)}$, the encoder model $\mathbf{E}$ is actually not involved in the proposed RDO process, while decoder model $\mathbf{D}$ is used once per iteration. The complete process of the proposed latent space

RDO at the frame-based is implemented by iteratively running the above steps. Algorithm 1 describes this process, in which a learning rate decay is also applied for an improved convergence. This technique takes two parameters of initial learning rate $\eta^{(0)}$ and decay rate β and is applied as follows:

$$\eta^{(i)} = \eta^{(0)}/(1 + \beta.i) \tag{10}$$

Algorithm 1 Iterative frame-level fine-tuning RDO

input: $x, \lambda, \Theta^*, \mathbf{E}, \mathbf{D}, \mathbf{C}$
parameters: $\eta^{(0)}, N, \beta$
output: z^*

$z^{(0)} \leftarrow \mathbf{E}(\mathbf{x}; \mathbf{\Theta}^*)$ (Eq. 7)

for $i := 0$ to $N-1$ **do**
  $\eta^{(i)} \leftarrow \eta^{(0)}/(1+\beta.i)$ (Eq. 10)
  $\hat{x}^{(i)} \leftarrow \mathbf{D}(z^{(i)}; \Theta^*)$ (Eq. 2)
  $\mathbf{r}^{(i)} \leftarrow \mathbf{EC}(z^{(i)})$ (Eq. 3)
  $\mathbf{d}^{(i)} \leftarrow ||x - \hat{x}^{(i)}||^2$ (Eq. 4)
  $\mathcal{L}_\lambda(z^{(i)}; \Theta^*) \leftarrow \mathbf{d}^{(i)} + \lambda \mathbf{r}^{(i)}$ (Eq. 8)
  $z^{(i+1)} \leftarrow z^{(i)} - \eta^{(i)} \nabla_z \mathcal{L}(z^{(i)}; \Theta^*)$ (Eq. 9)
end for

$z^* \leftarrow z^{(N-1)}$

GOP-based RDO with latent fine-tuning

Dependencies due to temporal frame referencing defined by the Group of Pictures (GOP) introduce a cascading effect of frame level decision changes. In particular, changing the current frame x_c can impact next frames in the GOP. To exploit this characteristic, the proposed method takes into account the GOP structure in the cost calculation of I- and P-frames. To avoid the complexity of considering the entire GOP, one single next frame, denoted as x_n, is included in the RDO of x_c.

The goal of each iteration of the GOP-based RDO is to obtain an altered latent as $z_c^{(i+1)}$ such that it will reduce the overall cost of not only current frame x_c, but also its next frame x_n. In other words, each iteration described in the previous section is followed by a normal (*i.e.* without latent fine-tuning RDO) encoding pass on x_n in order to provide its latent representation Eq. 11. This additional latent representation allows estimating the cascade effect due to changes of z_c in current iteration.

$$z_n^{(i)} = \mathbf{E}(x_n; \Theta^* | z_c^{(i)}). \tag{11}$$

A parameter $\alpha \in [0, 0.5]$ is defined to control the GOP impact by weighting the involvement of x_n in the RDO decision of x_c. In particular, two extremes of this parameter indicate no involvement of x_n ($\alpha = 0$) and equal involvement of x_c and x_n ($\alpha = 0.5$). Given α, we incorporate the GOP impact in the fine-tuning RDO process by changing the rate and distortion computation, as expressed in Eq. 12 and Eq. 13.

$$\mathbf{d}^G(z_c, z_n) = (1-\alpha)\mathbf{d}(x_c, z_c) + \alpha \mathbf{d}(x_n, z_n) \tag{12}$$

$$\mathbf{r}^G(z_c, z_n) = \alpha \mathbf{r}(z_c) + (1-\alpha)\mathbf{r}(z_n) \tag{13}$$

By also computing the distortion of the decoded signal of $z^{(i)}$ as $\hat{x}^{(i)}$, one can compute the approximate rate-distortion cost:

$$\mathcal{L}_\lambda^G(z_c^{(i)}, z_n^{(i)}; \Theta^*) = \mathbf{d}^G(z_c^{(i)}, z_n^{(i)}) + \lambda \mathbf{r}^G(z_c^{(i)}, z_n^{(i)})). \tag{14}$$

Finally, the latent fine-tuning of the GOP-based RDO is performed by back-propagating the gradient of the GOP loss with respect to only z_c, as expressed in $\nabla_{z_c} \mathcal{L}_\lambda^G$. It is important to note that, as indicated by the left side of Eq. 15, the GOP-based RDO updates only the current latent z_c and not the future latent z_n. However, for this update, the cascade effect on the cost of the future latent is taken into account by including z_n in computation of $\mathcal{L}_\lambda^G$.

$$z_c^{(i+1)} = z_c^{(i)} - \eta \nabla_{z_c} \mathcal{L}_\lambda^G(z_c^{(i)}, z_n^{(i)}; \Theta^*) \tag{15}$$

IV. EXPERIMENTS

Settings

The latent space RDO method has been implemented on top of an open-source[1] end-to-end LVC framework developed by Orange, called Artificial Intelligence for Video Coding (AIVC) [18]. It is noteworthy that this framework has won the Challenge on Learned Image Compression (CLIC) challenge in 2021. Therefore, in terms of compression performance, we aim at improving a codec which is already highly efficient. A set of six codec models are provided with the AIVC software, which are all optimized with a Lagrangian rate-distortion cost, as described in Section III.

The GOP impact parameter α, and the learning rate decay parameters $\eta^{(0)}$ and β, are the main parameters to be tuned in the experiments of this paper. All these parameters are chosen empirically, where the value of α was set to 0.4 and the learning rate decay parameters were set to $\eta^{(0)} = 5e^{-4}$ and $\beta = 0.5$.

The actual iterative fine-tuning algorithm was implemented in a conditional manner, such that the iterations would stop if the progress of the optimization is smaller than a given threshold. This threshold was chosen manually so that the trade-off between the encoding complexity and coding performance would be reasonable.

A GOP size of 16 was used in experiments, consisting of one I frame, eight P frames and seven B frames. As the GOP-based implementation imposes inevitable additional complexity due to encoding of next frame, we limited it to only two iterations. Precisely, each GOP-based experiment consists of two passes: a conditional frame-based latent fine-tuning (as described above), followed by two additional iterations of the GOP-based latent fine-tuning.

Two main metrics of have been used for performance assessment, namely, bitrate saving in terms of Bjøntegaard-Delta Bitrate (BD-BR) and complexity in terms of encoding time. For both metrics, the base AIVC coder without the proposed fine-tuning RDO is used as the anchor. Thus, the anchor has a performance of 0% in terms of BD-BR and 100% in terms of encoding time. Consequently, negative values of BD-BR indicate percentage of bitrate saving at the same level of PSNR quality and complexity values larger than 100% indicate the ratio of encoding time increase.

[1]https://github.com/Orange-OpenSource/AIVC

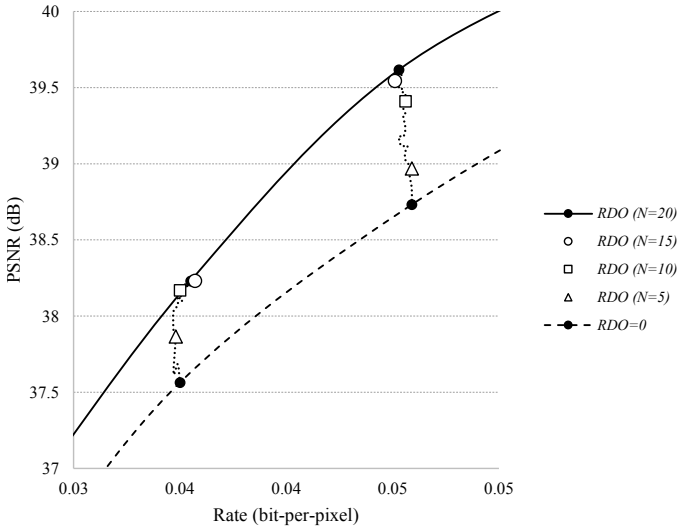

Fig. 2: Evolution of rate and PSNR through 20 iterations of the proposed RDO.

Performance

Table I compares the compression efficiency performance of the proposed latent fine-tuning RDO method in terms of BD-BR. In this table, two coding modes of All-Intra (AI) and Random-Access (RA) are considered, where for the RA mode, results of the frame-based and GOP-based are presented separately. As an early stop method is implemented in the iterative algorithm, the number of iterations for obtaining the results in Table I are different. This number varies between 7 and 15 iterations, where higher resolution sequences slightly tend to need more iterations before convergence. Moreover, as higher bitrates usually result in more non-zero latents, it was also observed that they require more iterations to converge. Fig. 2 shows the evolution of performance through twenty iterations of the proposed latent fine-tuning in All-Intra (AI) mode, with checkpoints at $N = 5, 10, 15$.

When comparing the two frame-based columns, it can be seen that the performance-complexity trade-off of the AI (with 10.6% gain for 364% encoding complexity) mode is better than the RA mode (with 9.4% gain for 470% encoding complexity). This is justified by the underlying base encoder, in which latents of an inter frame consists of both intra and inter prediction information that are conditionally coded, while the latents of an intra frame only include intra prediction information [18]. Therefore, fine-tuning of intra latents is naturally a simpler task.

When comparing the two RA columns, it is observed that the GOP-based method outperforms the frame-based RA method in terms of BD-BR and at the cost of additional complexity. Precisely, the GOP-based method brings about 2.2% additional BD-BR gain at the cost of about 75% additional complexity. One might argue that the frame-based method could have brought the same gain if it was not conditionally stopped using the performance threshold. However, it was observed that by allowing several more iterations of the frame-based method and even by manually tuning the learning rate, the achievable compression efficiency performance is still below the GOP-based method. To demonstrate the performance in different ranges of bitrate, Fig 3 shows Rate-PSNR curves. This figure is produced by applying the proposed GOP-based latent fine-tuning in Random-Access (RA) coding mode and on four 1080p sequences. As can be seen, the performance improvement due the proposed method is consistent at different bitrates.

Table I: Coding performance of the proposed method (RDO=1) in AI and RA (frame-based and GOP-based), in terms of BD-BR and encoding time, with respect to the base coder as anchor (RDO=0).

Class	Frame-based AI		Frame-based RA		GOP-based RA (α=0.4)	
	BD-BR	ET	BD-BR	ET	BD-BR	ET
A1	-10.8%	322%	-9.8%	454%	-11.4%	827%
A2	-10.6%	389%	-9.6%	419%	-11.6%	880%
B	-10.7%	356%	-9.7%	398%	-12.0%	797%
C	-11.0%	310%	-9.0%	510%	-11.4%	818%
D	-10.2%	395%	-9.2%	480%	-11.5%	804%
E	-10.2%	410%	-9.2%	560%	-11.5%	774%
All	**-10.6%**	**364%**	**-9.4%**	**470%**	**-11.6%**	**817%**

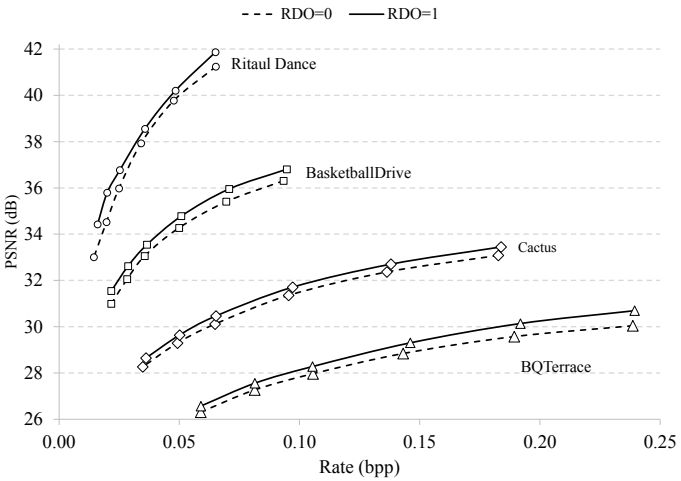

Fig. 3: RD curves of the proposed RDO method at different bitrate ranges of five 1920×1080 resolution sequences. Dashed and solid lines correspond to anchor and proposed methods, respectively.

V. CONCLUSION

In this paper, a latent domain RDO method is proposed that can be applied on top of any arbitrary learned-based end-to-end image and video coder. This method compensates the lack of flexibility in end-to-end encoders, in which the compression process is typically an inelastic mapping from the pixel-domain to the latent domain, performed by the layers of the encoder model. Moreover, the GOP-based version of the proposed method takes into account the temporal dependency of frames when optimizing the latent representation of a given frame. Experiments show that the both frame-based and GOP-based version of the proposed method can significantly improve the rate-distortion performance of the underlying with different complexity-performance trade-offs.

VI. REFERENCES

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

[2] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao, "DVC: An end-to-end deep video compression framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11006–11015.

[3] Abdelaziz Djelouah, Joaquim Campos, Simone Schaub-Meyer, and Christopher Schroers, "Neural inter-frame compression for video coding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6421–6429.

[4] Fabian Mentzer, Eirikur Agustsson, Johannes Ballé, David Minnen, Nick Johnston, and George Toderici, "Neural video compression using gans for detail synthesis and propagation," in *European Conference on Computer Vision*. Springer, 2022, pp. 562–578.

[5] Mohsen Abdoli, Felix Henry, Gordon Clare, Abderrahmane Jarmouni, and Kra-Tchimbie Koffi, "Spatial rate allocation for learning-based video coding," in *31st European Signal Processing Conference, EUSIPCO*, 2023.

[6] Yanghao Li, Xinyao Chen, Jisheng Li, Jiangtao Wen, Yuxing Han, Shan Liu, and Xiaozhong Xu, "Rate Control for Learned Video Compression," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 2829–2833.

[7] Fabian Brand, Kristian Fischer, and André Kaup, "Rate-Distortion Optimized Learning-Based Image Compression using an Adaptive Hierachical Autoencoder with Conditional Hyperprior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1885–1889.

[8] Fabian Brand, Kristian Fischer, Alexander Kopte, Marc Windsheimer, and André Kaup, "RDONet: Rate-Distortion Optimized Learned Image Compression With Variable Depth," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1759–1763.

[9] Fabian Brand, Kristian Fischer, Alexander Kopte, and André Kaup, "Learning True Rate-Distortion-Optimization for End-To-End Image Compression," *arXiv preprint arXiv:2201.01586*, 2022.

[10] Michael Schäfer, Sophie Pientka, Jonathan Pfaff, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand, "Rate-distortion-optimization for deep image compression," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 3737–3741.

[11] Yefei Wang, Dong Liu, Siwei Ma, Feng Wu, and Wen Gao, "Ensemble learning-based rate-distortion optimization for end-to-end image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 3, pp. 1193–1207, 2020.

[12] Guo Lu, Chunlei Cai, Xiaoyun Zhang, Li Chen, Wanli Ouyang, Dong Xu, and Zhiyong Gao, "Content adaptive and error propagation aware deep video compression," in *European Conference on Computer Vision*. Springer, 2020, pp. 456–472.

[13] Nam Le, Honglei Zhang, Francesco Cricri, Ramin Ghaznavi-Youvalari, Hamed Rezazadegan Tavakoli, and Esa Rahtu, "Learned image coding for machines: A content-adaptive approach," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[14] Xiao Wang, Wei Jiang, Wei Wang, Shan Liu, Brian Kulis, and Peter Chin, "Substitutional neural image compression," *arXiv preprint arXiv:2105.07512*, 2021.

[15] Wei Jiang, Wei Wang, Songnan Li, and Shan Liu, "Online Meta Adaptation for Variable-Rate Learned Image Compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 498–506.

[16] Joaquim Campos, Simon Meierhans, Abdelaziz Djelouah, and Christopher Schroers, "Content Adaptive Optimization for Neural Image Compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[17] Jing Zhao, Bin Li, Jiahao Li, Ruiqin Xiong, and Yan Lu, "A Universal Encoder Rate Distortion Optimization Framework for Learned Compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1880–1884.

[18] Théo Ladune, Gordon Clare, Pierrick Philippe, and Félix Henry, "Artificial Intelligence based Video Codec (AIVC) for CLIC 2022," in *CLIC 2022, 5th Workshop and Challenge on Learned Image Compression, CVPR 2022*, 2022.