Reducing The Mismatch Between Marginal and Learned Distributions in Neural Video Compression

Muhammet Balcilar
InterDigital, Inc.
Rennes, France
muhammet.balcilar@interdigital.com

Bharath Bhushan Damodaran
InterDigital, Inc.
Rennes, France
bharath.damodaran@interdigital.com

Pierre Hellier
InterDigital, Inc.
Rennes, France
pierre.hellier@interdigital.com

Abstract—During the last four years, we have witnessed the success of end-to-end trainable models for image compression. Compared to decades of incremental work, these machine learning (ML) techniques learn all the components of the compression technique, which explains their actual superiority. However, end-to-end ML models have not yet reached the performance of traditional video codecs such as VVC. Possible explanations can be put forward: lack of data to account for the temporal redundancy, or inefficiency of latent’s density estimation in the neural model. The latter problem can be defined by the discrepancy between the latent’s marginal distribution and the learned prior distribution. This mismatch, known as amortization gap of entropy model, enlarges the file size of compressed data. In this paper, we propose to evaluate the amortization gap for three state-of-the-art ML video compression methods. Second, we propose an efficient and generic method to solve the amortization gap and show that it leads to an improvement between 2% to 5% without impacting reconstruction quality.

Keywords—Neural video compression, Entropy model, Reparameterization.

I. INTRODUCTION

Image and video compression is a fundamental task in image processing, which has become crucial in the time of pandemic and increasing video streaming. Thanks to the community’s huge efforts for decades, traditional methods (including linear transformations under heavily optimized handcrafted techniques) have reached current state of the art rate-distortion (RD) performance and dominate current industrial codecs solutions. Alternatively, end-to-end trainable deep models have recently emerged, with promising results. Even though these methods clearly exceed many traditional techniques and surpass human capability for some tasks for a few years back very recently they beat the best traditional compressing method (VVC, versatile video coding [1]) even in terms of peak signal-to-noise ratio (PSNR) for single image compression [2]. However, their performance on video compression are still far from VVC, just on par with one generation back of the traditional method (HEVC, high efficiency video coding [3]). In addition to the inefficiency of neural models on capturing temporal redundancy, the mismatch between test latent’s normalized histograms and learned distributions in the entropy models may be a contributing factor.

Lossy image compression via end-to-end trainable models is special kind of Variational Autoencoder (VAE) that learns the transformations among data and latent codes and the probability models of these latent codes jointly [2], [4]–[7]. The problem is a multi-objective optimization problem where the model should be optimized for reconstruction quality and cross entropy of latent code w.r.t learned probabilities known as RD loss function. These neural image codecs were extended by using two VAEs, one for encoding motion information, another for encoding residual information in end-to-end video compression [8]–[12]. As all trainable models suffer from amortization gap [13] (may be optimal for entire dataset, but sub-optimal for given single test instance), neural compression models have a similar issue and this gap reduces the performance by either enlarging the file size or decreasing the reconstruction quality. The first fold solution to this problem is to apply post training for a given single test image/video, where some of them train just the encoder component of VAE [14], [15] in order to prevent extra signaling cost. The other selection is to fine tune all parts of the model by adding signaling cost to the loss function [16]. Second fold solutions do not apply time-consuming post training, but adjust some parameters of the model. For instance, in [17], only the entropy model’s amortization gap in end-to-end image compression is targeted and instance specific reparameterization of the latent distribution was proposed.

This paper is an extension of our previous work [17] to the video compression with some important differences. First, we introduce a general framework that generalizes end-to-end trainable models on video compression. Second, we analyse the amortization gap of all entropy models for different frames (I, B and P frames), for different information (motion and residual) in three recent neural video compression methods. Third, we identify the origin of the main performance drop and we show how it can be fixed. Last, but not least, we show the efficiency of our probability reparameterization method where the new parameters are kept into file considering the temporal redundancy of these parameters. According to the result, we decrease the file size of video between 2% to 5% in average without any effect on reconstruction quality. To the best of our knowledge, it is the first research on closing the amortization gap of neural video compression without post-training.

II. END-TO-END VIDEO COMPRESSION

Although the first end-to-end image compression model used only factorized entropy model in [4], following papers with hierarchical VAE based hyperprior entropy models as in [2], [5]–[7] became standard in neural image compression.
and now form the backbone of sota neural video compression. Let \( x, x, \hat{x}, \tilde{x} \) respectively denote the current image, motion warped current image, the reconstruction of current image and the reconstruction of reference image. Let us note \( v \) as the predicted motion information, \( y_m, \hat{y}_m, \hat{y}_r, y_r \) continuous latent, quantized (or noise added) latent of motion information and residual information respectively. Similarly, we will note \( z_m, \hat{z}_m, z_r, \hat{z}_r \) as the continuous side latent and quantized (or noise added) side latent of motion information and residual information respectively. Q(.) element-wise function applies quantization in test time or its continuous relaxation in train time as \( Q(x) = x + \epsilon \) that \( \epsilon \sim U(-0.5, 0.5) \) and \( W(\cdot) \) warsps given image w.r.t given motion. First VAE whose aim is to encode motion information takes current frame and reconstructed reference frame (or in B frame encoding, 2 reference frames) as inputs and finds warped image by \( y_m = gm_a(x, \hat{x}_r; \phi_m) \), \( \hat{y}_m = Q(y_m) \), \( v = gm_a(\hat{y}_m; \theta_m) \), \( \hat{z}_m = hm_a(y_m; \Phi_m) \), \( \hat{z}_m = Q(z_m) \), and \( \hat{x} = W(\hat{x}_r, v) \). Second VAE is for encoding residual information (difference between current frame and its motion warped version) and reconstruction of the image by applying \( y_r = gr_a(x - \hat{x}; \phi_r) \), \( \hat{y}_r = Q(y_r) \), \( z_r = hr_a(y_r; \theta_r) \), \( \hat{z}_r = Q(z_r) \) and \( \hat{x} = gr_a(\hat{y}_r; \theta_r) + \hat{x} \). As long as \( gm_a, gm_s, hm_a, gr_a, gr_s, hr_a \) are trainable deep models, neural video compression model can be written in followings;

\[
\mathcal{L} = E_{x, \hat{x}, r \sim p_k} \left[ -\log(p_f^{(m)}(\hat{y}_m | z_m, \Theta_m)) - \log(p_f^{(r)}(\hat{y}_r | z_r, \Theta_r)) - \log(p_{\hat{z}m}^{(m)}(\hat{z}_m | \Phi_m)) - \log(p_{\hat{z}r}^{(r)}(\hat{z}_r | \Theta_r)) + \lambda d(x, \hat{x}) \right],
\]

where \( p_f^{(m)}(.) | \Psi_m \) and \( p_f^{(r)}(.) | \Psi_r \) are factorized entropy model for motion and residual information, \( p_{\hat{z}m}^{(m)}(.) | \Phi_m \) and \( p_{\hat{z}r}^{(r)}(.) | \Theta_r \) are hyperprior entropy models for motion and residual information respectively implemented with neural networks, \( d(\cdot) \) is any distortion loss such as MSE for PSNR metric, \( \lambda \) hyperparameter that plays trade-off role between compression ratio and quality. Here the variables to be written into compressed file are motion’s main and side information \( (\hat{y}_m, \hat{z}_m) \) and residual’s main and side information \( (\hat{y}_r, \hat{z}_r) \) whose expected file sizes under learned entropy models are the first four parts of the loss in (1).

Factorized entropy models learn the probability mass function (pmf) of the symbols for each feature band of \( \hat{z}_m \) and \( \hat{z}_r \) separately. Thus, learned pmf values are defined by parameters of factorized entropy model \( \Psi_m, \Psi_r \). If the side latent of motion or residual information is \( \hat{z}_m, \hat{z}_r \) \( \in \mathbb{R}^{k \times k \times f} \) and \( \hat{p}_{\hat{z}m}^{(m)}(.) \) is the pmf table of \( s \)-th feature band for motion or residual information, the factorized entropy model applies;

\[
p_f^{(m|r)}(\hat{y}_m | \hat{z}_m; \Psi_{m,r}) = \prod_{i=1}^{k} \prod_{j=1}^{k} \hat{p}_{\hat{z}m}^{(m|z)}(\hat{z}_m[i,j,c]).
\]

Hyperprior entropy model learns the parameters of main information’s probability function (usually gaussian or laplacian distribution is used) using already encoded side latent by \( \mu, \sigma = hm_a(\hat{z}_m, \Phi_m) \) or \( \mu, \sigma = hr_a(\hat{z}_r, \Theta_r) \) for motion or residual information and applies as follows;

\[
p_h^{(m|r)}(\hat{y}_m | \hat{z}_m; \Theta_{m,r}) = \prod_{i=1 \in N(\sigma)} \hat{N}(\hat{y}_m[i,j]; 0, \sigma_c).
\]

where \( \hat{y}_{m,[r]} = Q(\hat{y}_{m,[r]} - \mu_c), \hat{y}_{m,[r]} = \hat{y}_{m,[r]} + \mu_c, \sigma_c \) is \( c \)-th predefined scale, \( \hat{N}(\cdot; \mu, \sigma) \) is a set of latent index whose winning scale is \( \sigma, \hat{N}(x; \mu_c, \sigma_c) \) pmf value of \( x \) for 1-d gaussian distribution with \( \mu, \sigma \) parameters, \( s \) is number of predefined scale values of gaussian distribution and \( hm_a, hr_a \) are another deep neural networks.

In this framework, end-to-end video compression consists of 5 trainable components for motion information \( gm_a, gm_s, hm_a, hm_s, p_f^{(m)} \) and 5 components for residual information \( gr_a, gr_s, hr_a, hr_s, p_f^{(r)} \) parameterized by \( \phi_m, \theta_m, \Phi_m, \Psi_m, \Phi_r, \Theta_r \) and \( \phi_r, \theta_r, \Phi_r, \Theta_r, \Psi_r \) respectively and a non-trainable motion warping function \( W(\cdot, \cdot) \). The selection of all these 11 components explains the differences between studied end-to-end video compression methods.

III. METHOD

The gap caused by the mismatch between marginal and learned distribution and our proposals are given in this section.

A. Gap of the Entropy Models

When considering intra frame encoding (I frame, as single image encoding) there is no motion nor reference images, thus first two component of the loss in (1) is canceled out and \( \hat{x} = 0 \). Consequently, only the residual’s main and side information have to be encoded for I frames. However, for inter frame encoding (either B or P), both motion’s and residual’s main and side information are encoded. Using entropy models in (2) and (3), we can write expected bitlength of \( c \)-th feature band of side information as follows:

\[
l_{f,c}^{(m|r)} = - \sum_{i=1}^{k} \sum_{j=1}^{k} \log \left( \hat{p}_{\hat{z}m}^{(m|z)}(\hat{z}_m[i,j,c]) \right).
\]

We can write expected bitlength of main information that represented by \( c \)-th predefined scale as follows;

\[
l_{h,c}^{(m|r)} = - \sum_{i \in N(\sigma_c)} \log \left( \hat{N}(\hat{y}_m[i,j]; 0, \sigma_c) \right).
\]

Considering all side information’s bitlength, it can be calculated by \( l_{f,c}^{(m|r)} = \sum_{i} l_{f,c}^{(m|r)} \) and \( l_{h,c}^{(m|r)} = \sum_{i} l_{h,c}^{(m|r)} \) for all main information’s bitlength. This leads us to calculate baseline method’s bitlength of inter frame by summing up all 4 informations’ bitlength by \( l(b) = l_{f}^{(m)} + l_{f}^{(r)} + l_{h}^{(m)} + l_{h}^{(r)} \).

The optimality of bit length of each information depends on how learned pmfs (\( \hat{p}_{\hat{z}m}, \hat{p}_{\hat{z}r}, \hat{N}(\hat{y}_m,0,\sigma), \hat{N}(\hat{y}_r,0,\sigma) \) are closed to marginal distribution of latents (\( \hat{z}_m, \hat{z}_r, \hat{y}_m, \hat{y}_r \)) and it is defined as amortization gap of entropy models in neural image compression [17]. This mismatch can be seen by differences between green curves and blue histogram bars in Fig 1. Following the same procedure, we can calculate the theoretical limit of expected bit length of each information by simply replacing learned pmf by corresponding latent’s normalized histogram as follows;

\[
l_{f,c}^{(n|m|r)} = - \sum_{i=1}^{k} \sum_{j=1}^{k} \log \left( \hat{h}_{m,c}^{(m|z)}(\hat{z}_m[i,j,c]) \right).
\]
where $h^{(m)}_{f,c}(x) = \sum_{i \in \mathcal{N}(\sigma_c)} \delta(x, \hat{y}_{m|r,i})/k^2$ is normalized frequency of symbol $x$ on $k \times k$ slice of $\hat{y}_{m|r}$. Theoretical limit of expected bit length of main information can be written as follows:

$$t^{(m|r)}_{h,c} = - \sum_{i \in \mathcal{N}(\sigma_c)} \log \left( h^{(m)}_{h,c}(\hat{y}_{m|r,i}) \right),$$

(7)

where $h^{(m)}_{f,c}(x) = \sum_{i \in \mathcal{N}(\sigma_c)} \delta(x, \hat{y}_{m|r,i})/|\mathcal{N}(\sigma_c)|$ is normalized frequency of symbol $x$ on $\hat{y}_{m|r,i}$ where $i \in \mathcal{N}(\sigma_c)$ and $|\mathcal{N}(\sigma_c)|$ is the number of element in the given set. Following the same procedure, bitlength of all side and main information can be calculated by $t^{(m|r)}_{f,c} = \sum_{c \in \mathcal{C}} t^{(m|r)}_{f,c}$ and $t^{(m|r)}_{h} = \sum_{c \in \mathcal{C}} t^{(m|r)}_{h,c}$.

Thus, theoretical limit of bitlength of inter frame would be $t = t^{(m)} + t^{(m)} + t^{(r)} + t^{(r)}$. The differences between baseline model’s each information’s bitlength and theoretical limits of the bitlength gives the amortization gap of corresponding type of information.

B. Temporal Re-parameterization

In order to reduce the mismatch, learned pmfs in the model were replaced by some parametric distribution $\tilde{p}_{f|h}(.; \beta)$ whose parameters $\beta$, to be signalled to the receiver, are obtained by fitting the actual histogram of the latents as in [17]. We use the same reparameterizations that are truncated Gaussian mixture for factorized entropy (motion/residual’s side information) and truncated zero-mean Gaussian distribution for hyperprior entropy (motion/residual’s main information) where the parameters are discretized into 10 bits. Sample probabilities with $K = 1$ Gaussian mixture and zero-mean Gaussian can be found in Fig. 1 in the red curves for all four types of information. When $t^{(m|r)}_{f|h}$ is the bitlength of motion/residual’s side/main information under $\tilde{p}_{f|h}(.; \beta)$ probabilities, plus the signaling cost of $\beta$, the bitlength of interface in our proposal would be $t^{(o)} = t^{(m)} + t^{(m)} + t^{(r)} + t^{(r)}$.

To take temporary redundancies of parameter signaling cost into account, we use $S$-bit temporal mask to explain if the previously encoded interface’s corresponding parameter is the same or not for the top $S$ number of pmf tables. If this bit is 1, there is no need to encode new parameters but uses previously encoded interface’s parameters. In this way, we decrease the temporal redundancy of these parameters. When considering the signaling cost of each pmf table, we have seen that just a few of the pmf table is worth to replace with reparameterization one but vast majority of them is not. To determine which is worth or not, we reorder the pmf table w.r.t its own entropy (in hyperprior model, it is not necessary since the pmf tables are ordered by predefined scale values and lower scales pmf table highly likely carries more information).

Later we test top $S$ number of pmf table one by one if the expected bit gain is larger than reparameterization cost or not. The new parameters are written to the file and pmf is replaced if the gain is larger than that signaling cost. When $K$ number of mixture is used, the signaling cost is $10(3K - 1)$ bits in factorized entropy. Since there is just one parameter in zero-mean gaussian, the signaling cost is 10 bits in hyperprior model when temporal mask is 0. If the parameters are the same with previously encoded interface, signaling cost is 0 bit for all entropy models. To tell the receiver which pmf is replaced, we need to use 1-bit replacement mask in addition to replace pmf’s parameters and temporal mask.

Algorithm 1 Encoding with Factorized Entropy

**Input:** Learned pmfs $\tilde{p}_c^{(c)}$ for $c = 1, \ldots, s$ w.r.t its entropy

**Output:** latent bitstream $lb$ and parameter bitstream $pb$.

**Recoder $\tilde{p}_c^{(c)}$ for $c = 1, \ldots, s$ w.r.t its entropy**

for $c = 1$ to $S$ do

$\beta^{(c)} = \arg \max_{\beta} \sum_{i,j} \log \left( \tilde{p}(\hat{z}_{m,r}; \beta^{(c)}) \right)$ find parameters

$\beta^{(c)} = Q(\beta^{(c)})$ quantize the best parameters

$G = \sum_{i,j} \log \left( \tilde{p}(\hat{z}_{m,r}; \beta^{(c)}) \right)$ goal gain

if $\beta^{(c)}$ is None and $G < 30K - 10$ then

```
ph.write(True) // write temporal bit
\beta^{(c)} = None // set best params
```

else if $\beta^{(c)}$ is $\tilde{\beta}^{(c)}$ and $G > 0$ then

```
ph.write(True) // write temporal bit
lb.write(\tilde{\beta}^{(c)}) // write latent w.r.t learned pmf
```

else if $G > 30K - 10$ then

```
ph.write(False) // write temporal bit
ph.write(\tilde{\beta}^{(c)}) // write replacement bit
```

else

```
ph.write(False) // write temporal bit
ph.write(\tilde{\beta}^{(c)}) // write replacement bit
```

end if

end for

for $c = S + 1$ to $s$ do

```
lb.write(\tilde{\beta}^{(c)}) // write latent w.r.t learned pmf
```

end for
encoded interframe’s parameter is $\hat{\beta}_p^{(c)}$ (all params of the first interframe, and the pmf table which are not replaced are None), detailed flow of factorized entropy encoding is given in Algorithm 1. Extending it on hyperprior entropy model is straightforward by changing only the variable names/indexing.

IV. Results

In order to test our proposal, we use 7 video sequences in UVG dataset [18] at 1080p resolution. We use 10 set of consecutive 16 frames (total 160 frames) of each sequence and compress them by compressai [19] implementation of SSF [9], and author’s official implementations of LHBDC [11] and AIVC [10]. Since all methods were proposed on different purpose, we do not make comparison of these methods but measure saving of our generic solution up to these baselines (mainly motion information represents the vast majority of the bitstream (76.9% – 93.8%) whose gap is the smallest (1.3% – 4.3%). Our proposal can fill almost half of this gap. Residual’s side information is the second largest information type in the interframe’s bitstream where the biggest gap exists (10.6% – 43.9%). Truncated gaussian mixture fills more than half of these gaps. Even though the gap of motion information is larger, the amount of the data is small and its effect on general performance is limited. Since our reparameterization approach has fixed cost, if the amount of data is very low (less than 1% in AIVC and around 2% in SSF’s main motion information and less than 1% in LHBDC’s side motion information), proposed method cannot save any bits or saving is very limited even though the gap is 10% – 20%. Our performance on side information (factorized entropy) is much more larger than performance on main information (hyperprior entropy) for all methods. In average of 16 frame length video for the lowest bpp objective, the gap is between 4.1% – 7.7% and our saving is between 2.0% – 4.7%. To measure the gap and our savings in different quality/rate, we perform several tests with the pretrained models provided by authors. Results are given in Fig 2 and show that gap and our savings decreases with the increment of quality/rate. Even at the highest quality/rate objective, the gain is better than 1% for all methods.

In all models and frame types, the residual’s main information represents the vast majority of the bitstream (76.9% – 93.8%) whose gap is the smallest (1.3% – 4.3%). Our proposal can fill almost half of this gap. Residual’s side information is the second largest information type in the interframe’s bitstream where the biggest gap exists (10.6% – 43.9%). Truncated gaussian mixture fills more than half of these gaps. Even though the gap of motion information is larger, the amount of the data is small and its effect on general performance is limited. Since our reparameterization approach has fixed cost, if the amount of data is very low (less than 1% in AIVC and around 2% in SSF’s main motion information and less than 1% in LHBDC’s side motion information), proposed method cannot save any bits or saving is very limited even though the gap is 10% – 20%. Our performance on side information (factorized entropy) is much more larger than performance on main information (hyperprior entropy) for all methods. In average of 16 frame length video for the lowest bpp objective, the gap is between 4.1% – 7.7% and our saving is between 2.0% – 4.7%. To measure the gap and our savings in different quality/rate, we perform several tests with the pretrained models provided by authors. Results are given in Fig 2 and show that gap and our savings decreases with the increment of quality/rate. Even at the highest quality/rate objective, the gain is better than 1% for all methods.

V. Conclusions

In this work, we propose a low computational demanding solution for the inefficiency of entropy coding in neural video compression. Since the method is generic, it can be applied up to any neural video compression and decrease the bitlength between 2% – 5% according to our tests. The amortization gap is inevitable in all ML based models. However, the connection of the amount of this gap to the certain type of neural codec should be explored in order to develop better codec which is inside our agenda as a future work.
REFERENCES

[1] B. Bross, J. Chen, J.-R. Ohm, G. J. Sullivan, and Y.-K. Wang, “Developments in international video coding standardization after avc, with an overview of versatile video coding (vvc),” *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1463–1493, 2021.

[2] G. Gao, P. You, R. Pan, S. Han, Y. Zhang, Y. Dai, and H. Lee, “Neural image compression via attentional multi-scale back projection and frequency decomposition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 677–14 686.

[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (hevc) standard,” *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[4] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression,” in *ICLR*, 2017.

[5] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, “Variational image compression with a scale hyperprior,” in *ICLR*, 2018.

[6] D. Minnen, J. Ballé, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, 2018.

[7] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Learned image compression with discretized gaussian mixture likelihoods and attention modules,” in *CVPR*, 2020.

[8] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, “Dvc: An end-to-end deep video compression framework,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 006–11 015.

[9] E. Agustsson, D. Minnen, N. Johnston, J. Balle, S. J. Hwang, and G. Toderici, “Scale-space flow for end-to-end optimized video compression,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8503–8512.

[10] T. Ladune and P. Philippe, “Aivc: Artificial intelligence based video codec,” *arXiv preprint arXiv:2202.04365*, 2022.

[11] M. A. Yılmaz and A. M. Tekalp, “End-to-end rate-distortion optimized learned hierarchical bi-directional video compression,” *IEEE Transactions on Image Processing*, 2021.

[12] R. Pourreza and T. Cohen, “Extending neural p-frame codecs for b-frame coding,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6680–6689.

[13] C. Cremer, X. Li, and D. Duvenaud, “Inference suboptimality in variational autoencoders,” *ICML*, 2018.

[14] G. Lu, C. Cai, X. Zhang, L. Chen, W. Ouyang, D. Xu, and Z. Gao, “Content adaptive and error propagation aware deep video compression,” in *ECCV*, vol. 12347, 2020, pp. 456–472.

[15] J. Campos, S. Meierhans, A. Djelouah, and C. Schroers, “Content adaptive optimization for neural image compression,” in *CVPR Workshops*, June 2019.

[16] T. van Rozendaal, I. A. Huijben, and T. S. Cohen, “Overfitting for fun and profit: Instance-adaptive data compression,” in *ICLR*, 2021.

[17] M. Balcilar, B. Damodaran, and P. Hellier, “Reducing the amortization gap of entropy bottleneck in end-to-end image compression,” in *Picture Coding Symposium (PCS)*, 2022.

[18] A. Mercat, M. Viitanen, and J. Vanne, “Uvg dataset: 50/120fps 4k sequences for video codec analysis and development,” in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020, pp. 297–302.

[19] J. Bégaint, F. Racapé, S. Feltman, and A. Pushpapraja, “Compressai: a pytorch library and evaluation platform for end-to-end compression research,” *arXiv preprint arXiv:2011.03029*, 2020.