Improving Multi-generation Robustness of Learned Image Compression

Litian Li\textsuperscript{1}, Zheng Yang\textsuperscript{1}, Yongqi Zhai\textsuperscript{1,2}, Jiayu Yang\textsuperscript{1,2}, Ronggang Wang\textsuperscript{1,2*}

\textsuperscript{1} Shenzhen Graduate School, Peking University
\textsuperscript{2} Peng Cheng Laboratory, Shenzhen, China
{lilitian, yang, zhaiyongqi}@stu.pku.edu.cn, jiayuyang@pku.edu.cn, rgwang@pkusz.edu.cn

Abstract—Benefiting from flexible network designs and end-to-end joint optimization approach, learned image compression (LIC) has demonstrated excellent coding performance in recent years. However, existing compression models suffer from serious multi-generation loss, which always occurs during image editing and transcoding. During the process of repeatedly encoding and decoding, the image quality will rapidly degrade, resulting in various types of distortion, which significantly limits the practical application of LIC. In this paper, a thorough analysis is carried out to determine the source of generative loss in successive image compression (SIC). We point out and solve the quantization drift problem that affects SIC, reversibility loss function as well as channel relaxation method are proposed to further reduce the generation loss. Experiments show that by using our proposed solutions, LIC can achieve comparable performance to BPG even after 50 times reencoding. Our code is available at https://github.com/leelitian/Multi-Generation-Robust-Coding.

Index Terms—image compression, multi-generation robustness

I. INTRODUCTION

Image compression is one of the most fundamental technologies in the multimedia field. Efficient image compression technology, especially lossy compression technology, provides support for the storage and transmission of massive image data. Traditional coding standards, such as JPEG [1], BPG [2], and the latest VVC [3], rely on hand-crafted modules to remove spatial and statistical redundancy in image data to achieve the purpose of compression. They typically follow a hybrid prediction, transformation, quantization, and entropy coding framework, and utilize a series of well-designed coding tools to improve coding efficiency.

In recent years, image coding methods based on end-to-end optimization have been rapidly explored and developed, and show promise to become the next-generation coding standard. On the one hand, with the powerful image understanding and generation capabilities of deep learning, some very recent works outperform VVC in PSNR and MS-SSIM [4]–[9]. On the other hand, to meet the needs of industrial applications, researchers have designed flexible modules to implement variable bitrate [10] and scalable coding [11]. For HDR image coding [12], stereo image coding [13], [14], and omnidirectional image coding [15], learning-based compression model also showed superiority over traditional codecs.

Despite the remarkable progress of deep learning in the image compression field, two problems need to be solved before it comes into our lives. One is the model lightweight, mainly to reduce the network complexity, which is crucial for mobile devices [16]. Another is the model robustness, which aims to make the model more reliable. In this paper, we focus on improving the stability of the models during successive compression, called multi-generation robustness.

Multi-generation coding is a repeated compression and decompression process of images or videos [17], which often occurs in multimedia application scenarios such as editing, transcoding, and redistribution. In the case of lossless compression, the repeatedly encoded and decoded image is identical to the original image. For lossy compression methods, the first compression will induce distortion, but the distortion will ideally not increase when the decoded image is reencoded with the same configuration. However, the instability of successive deep image compression is discovered in previous work [18]. After an image is repeatedly compressed by an existing compression model, the quality will drastically degrade, resulting in blurring and color casts. Moreover, compression artifacts can accumulate in multiple decompression-re-compression cycles and eventually corrupt the image. The salient patterns not only degrade the visual experience, but the induced high-frequency noise can significantly increase the bitrate of the image, which is particularly detrimental to practical multimedia applications.

This paper aims to thoroughly analyze the factors that affect the multi-generation robustness of LIC and to enhance the stability of SIC without impacting the rate-distortion performance of the first encoding. According to our analysis, the quantization strategy, and the reversibility of transformation play key roles in multi-generation robustness. To reduce error accumulation in SIC cycles, we suggest using straight quantization (SQ) instead of corrected quantization. To increase the reversibility of transformation, reversibility loss (RL) and channel relaxation (CR) methods are proposed. Experiments show that the robustness of the model enhanced by our solutions significantly outperforms previous work [18], and our methods can be easily extended to different learned compression models. Fig. 1 shows the extraordinary performance of our two solutions.
II. RELATED WORKS

A. Learned Variational Image Compression

Recent lossy image compression frameworks are based on transform coding [20], where the encoder applies an analysis transform $g_a$ mapping the input image $x$ to its latent representation $y$. The latent representation $y$ is quantized by $Q(\cdot)$ and entropy coded. For reconstruction, compressed latent $\hat{y}$ obtained from entropy decoding is passed through synthesis transform $g_s$ to yield $\hat{x}$. Mathematically, we can write:

$$y = g_a(x); \quad \hat{y} = Q(y); \quad \hat{x} = g_s(\hat{y}).$$

(1)

To deal with the zero gradient problem caused by quantization, additive uniform noise is applied to $y$ as a continuous approximation during training with $\hat{y} = y + U(-0.5, 0.5)$ [21]. Then this approach is equivalent to a variational autoencoder. The framework is optimized in an end-to-end manner:

$$\mathcal{L} = R(\hat{y}) + AD(x, \hat{x}),$$

(2)

where the hyper-parameter $\lambda$ is used to realize the trade-off between the estimated bitrate $R$ and image reconstruction distortion $D$.

A key challenge in LIC is estimating the entropy of quantized latent $\hat{y}$, for more accurate models typically result in better performance. As shown in Figure 2, existing models usually adopt a joint hyperprior and context entropy model to obtain a more accurate probability estimation. Each symbol $y_i$ of the latent representation is usually modeled as a Gaussian distribution $N(\mu_i, \sigma_i^2)$. The hyperprior model [21] captures the global information in $y$ using the hyper analyzer $h_a$, and the resulting $\hat{z}$ is transmitted to the decoder as side information. The context model [19] aims to further save bits by exploiting the correlation between already decoded symbols $\hat{y}_{<i}$ and the currently decoding symbol $\hat{y}_i$. By jointly combining these two methods together, the entropy parameter of each symbol can be formulated as:

$$\Phi_i = (\mu_i, \sigma_i) = g_{ep}(h_s(\hat{z}), g_{en}(\hat{y}_{<i})).$$

(3)

where $h_s(\hat{z})$ denote the hyperprior feature and $g_{en}(\hat{y}_{<i})$ denote the context feature. The two features are fused by $g_{ep}$ before yield probability parameter $\Phi_i$.

B. Multi-generation Robust Coding

Multi-generation coding is a process of repeatedly compressing decoded pictures [22], and the resulting generation loss has a non-negligible impact on multimedia applications.

Multi-generation robustness for LIC was first discussed in Kim’s work [18]. A SIC benchmark was conducted for the state-of-the-art learning-based models, results show that most models suffer from serious quality loss during repeated compression. The definition of SIC is as follows:

$$f_{SIM} = RC \circ g_s \circ Q \circ g_a,$$

(4)

$$x_n = f_{SIM}^n(x_0),$$

(5)

where $\circ$ denotes the function composition. $f_{SIM}^n(\cdot)$ denotes one compression-decompression cycle, following a pipeline of analysis transform $g_a$, quantization $Q$, synthesis transform $g_s$, rounding and clipping $RC$. $x_0$ is the original image, and $x_n$ is the image after $n$ times SIC cycles.

To reduce the multi-generation loss, feature identity (FI) loss [18] is proposed to be added during training time, as illustrated in the equation:

$$\mathcal{L}_{FI} = ||\hat{y}_1 - \hat{y}_0||_2,$$

(6)

where $\hat{y}_0 = Q(g_a(x_0))$ and $\hat{y}_1 = Q(g_a(g_s(\hat{y}_0)))$. Although the degradation rate of image quality was slowed down by using the loss function, the degradation trend has not changed. The reason maybe they did not address the problem of quantization drift, which will be discussed in the next section.

III. WHAT IMPACTS ROBUSTNESS

A. Quantization Drift Problem

Quantization is an important module of image compression, and its implementation is crucial to the stability of SIC. In the ideal case of idempotent reencoding, the quantized
An entropy model with hyperprior and context, the output is from the entropy model. The whole quantization process can be expressed by the following formula, which we call “corrected quantization”:

\[ \hat{y}_i = \text{round}(y_i - \mu_i) + \mu_i, \]  

where the \( \mu_i \) is from the entropy model.

Recent work tends to follow the quantization implementation in [20]. When encoding the symbol \( y_i \), they first subtract its mean \( \mu_i \), followed by quantization and entropy encoding, and finally add \( \mu_i \). The latent representation \( \hat{y}_0 \) and \( \hat{y}_1 \) should be equal in two-round encoding, so that the decoded images are also the same.

Corrected quantization is widely used in the state-of-the-art works [6], [19], [21]. However, it can seriously impact the multi-generation robustness of LIC. As illustrated in Fig. 2, in an entropy model with hyperprior and context, the output \( \mu \) depends on the feature \( y \) and the already decoded symbol \( \hat{y}_{ci} \). Assume that \( y \) is slightly perturbed, \( \mu \) will change and lead change in \( \hat{y} \), and a chain reaction will further occur due to the context model. In this case, to make \( \hat{y}_0 \) and \( \hat{y}_1 \) equal in two-round encoding, the latent representation \( y_0 \) and \( y_1 \) must be the same. However, this is almost impossible due to the existence of quantization and clip errors in the SIC cycle. “mbt18” and “mbt18+FI” in Fig. 1 show that corrected quantization causes perturbations to accumulate during SIC, resulting in continuous degradation of image quality.

**IV. PROPOSED METHODS**

**A. Straight Quantization**

To address the quantization drift problem, we propose to use straight quantization instead of corrected quantization at testing time. In the process of straight quantization, rounding and entropy coding are performed directly on \( y_i \) instead of \( y_i - \mu_i \), using the following quantization process instead of the Equation (7):

\[ \hat{y}_i = \text{round}(y_i). \]  

The latent representation \( \hat{y} \) fed into the decoder is a discrete value, and the quantization process of each symbol \( y_i \) is relatively independent, no longer depends on the whole \( y \) and the already decoded \( \hat{y}_{ci} \). In the process of successive compression, as long as the values of \( y_0 \) and \( y_1 \) fall in the same quantization interval, the quantized value \( \hat{y}_0 \) and \( \hat{y}_1 \) will be the same and so does the decoded images.

To avoid train-test mismatch, we use the straight-through estimator (STE) [24] rather than the noisy approximation for training. STE applies hard rounding in the forward pass and uses the modified gradient in the backward pass, which we find effective for multi-generation robust coding. As for entropy rate estimation, we still follow the mainstream works and adopt the method of adding uniform noise.

**B. Reversibility Loss Function**

There is no explicit constraint on the reversibility of the transformation in the original loss function (Equation 2). To enhance the reversibility and reduce the generation loss, we add a reversible constraint to the original distortion term at training time, the new loss function can be written as:

\[ \mathcal{L} = R(\hat{y}) + \lambda(D(x, \hat{x}) + \alpha D(x, \bar{x})), \]  

where \( \bar{x} = g_s(y) \) and \( \alpha \) is a hyperparameter controlling a trade-off between the real distortion term and reversible constraint term. The approach is shown in Fig. 3. Different from feature identity loss proposed in [18], reversibility loss does not consider the quantization process. We directly send the latent \( y \) before quantization into the decoder and obtain the image \( \bar{x} \), and expect \( \bar{x} \) to be equal with the original image \( x \), thus constraining the reversibility of \( g_s \) and \( g_a \).

networks to construct nonlinear transform \( g_a(\cdot) \) and \( g_s(\cdot) \). As shown in Fig. 3, \( g_a(\cdot) \) generally contains 4 downsampling operations, the feature map of the middle layer contains \( N \) channels, while the latent representation contains \( M \) channels. A symmetric architecture is usually adopted by \( g_s(\cdot) \). The network parameters are optimized for the rate-distortion function, where the distortion term is \( d(x, g_a(\hat{y})) \) and \( \hat{y} \) is the quantized approximation of \( g_s(x) \). The distortion term does not explicitly constrain the reversibility of \( g_s(\cdot) \) and \( g_a(\cdot) \). And to obtain a more compact latent representation, limited \( M \) is set in previous works [19], [21], [23], which may also restrict the reversibility of the transform network.
C. Channel Relaxation

Existing LIC models use a transformation network $g_a(\cdot)$ with 4 spatial downsampling operations to transform the original image $x \in \mathbb{R}^{h \times w \times 3}$ into a compact representation $y \in \mathbb{R}^{M \times \frac{h}{16} \times \frac{w}{16}}$, where $h$ and $w$ denote the height and width of the original image, $M$ denotes the channel number of latent feature $y$. In previous works [19], [21], [23], $M$ is generally set to be larger in high-bitrate models. The reason may be that the latent with a larger dimension can retain more information at the entropy bottleneck, thus enabling the decoder to recover more details.

Our pre-experiments confirm that limited channels constrain the information-holding ability of the latent representation. Too much information is lost in transformation, resulting in weaker reversibility of the transform network. Dimensional relaxation of the compact latent $y$ can reduce the constraints, making information loss in the quantization process instead of transformation, which coincides with traditional codecs. The method will reduce the generation loss without changing the network architecture and optimization function. To implement channel relaxation, we obtain hyperparameter $M$ suitable for every bitrate through an enumerative algorithm. $M$ will affect the parameter dimensions of the layers close to the entropy bottleneck (e.g., input and output channel numbers of convolutional layers).

V. Experiment

In this section, we first prove the effectiveness of the straight quantization strategy and the reversibility enhancement strategy, followed by a detailed test and comparison of the two solutions for multi-generation robust coding.

A. Experiment Setup

Baseline Models. We choose the joint autoregressive and hierarchical entropy model “mbt18” [19] and its variant “cheng20” [23] as baseline models. Different from mbt18, cheng20 uses stacked residual blocks for analysis and synthesis transformation instead of vanilla convolutional layers. When evaluating the baseline models, we use the pre-trained weights from CompressAI [25] with reconstruction qualities ranging from 1 to 6.

Training details. We use Flicker dataset used in [6] for training. The images are cropped as 256 $\times$ 256 patches before input into the networks. All the models are trained for 1.8M steps with a batch size of 8 using Adam optimizer, with an initial learning rate of $10^{-4}$, and reduced to $10^{-5}$ for the last 0.2M steps. Models are optimized with MSE (mean square error) quality metric, which is consistent with traditional codecs. $\lambda$ is chosen from the set $\{0.0016, 0.0032, 0.0075, 0.015, 0.03, 0.045\}$.

Evaluation details. We evaluate learned models and traditional codecs on two commonly used datasets for image compression, which are Kodak [26] and CLIC Professional [27]. Consistent with [18], we set the number of SIC cycles to $n = 50$. We not only evaluate the loss of PSNR during successive compression, but also the rate-distortion performance at $n$-th time, where the distortion is computed by $D(x_0, x_n)$, and the rate is $R(x_{n-1})$.

Fig. 4. Evaluation of quantization strategies on Kodak.

B. Quantization Strategy

Quantization strategies play a crucial role in multi-generation robustness. The results on the Kodak dataset are shown in Fig 4. As shown in the figure, “mbt18+SQ” uses the proposed straight quantization strategy, while the other three
models use the default corrected quantization. After 50 times reencoding, “mbt18+SQ” still maintains good reconstruction quality, while other models show significant quality degradation. Even with the reversibility enhancement methods (CR or RL), the drift problem caused by corrected quantization cannot be eliminated, justifying the necessity of straight quantization in multi-generation robust coding. We also note that benefit from avoiding train-test mismatch, mbt18+SQ shows close rate-distortion performance to mbt18 in the first compression, suggesting that straight quantization is an excellent alternative to corrected quantization.

![Fig. 5. Effect of channel relaxation and reversibility loss. The numbers on the x-axis represent M, “320+RL” denotes M = 320 with reversibility loss.](image)

**TABLE I**

PROPOSED RELAXED M FOR MBT18.

| Quality | 1 | 2 | 3 | 4 | 5 | 6 |
|---------|---|---|---|---|---|---|
| Original Channel | 192 | 192 | 320 | 320 | |
| Channel Relaxation | 192 | 448 | 512 | 576 | |

**C. Reversibility Enhancement**

Fig. 4 shows that only using the straight quantization strategy, the generation loss is still large at high bit rates. According to our analysis, channel relaxation (CR) or reversibility loss (RL) can further improve the robustness of the model. To explore the effect of latent channel number M and the reversibility loss function, M is increased from 256 to 640 by 64 each time, α in Equation 9 is set to 1 empirically. We use $d(x_0, x_0)$ to characterize transform reversibility. To explore the impact of reversibility on generation robustness, we compute the first-generation loss as $d(x_2, x_0) - d(x_1, x_0)$. The experimental results on the mbt18 model improved by “SQ” with quality = 5 are shown in Fig. 5.

The results reveal that stronger reversibility leads to smaller generation loss. As the number of channels M increases, the reversibility shows a trend of first increasing and then decreasing. The reason is that a slightly increased M relaxes the information bottleneck, and reduces information loss in the transformation. While too large M will lead to unstable and insufficient training. Similar conclusions can be drawn at different bitrates and benchmark models. After adding explicit constraints in the loss function, the reversibility of the transformation is significantly increased and exhibits the smallest first-generation loss. Table I presents the M with the best robustness after channel relaxation for mbt18. Note that only the models with high bitrate are relaxed.

**D. Overall Evaluation**

Through pre-experiments, we finally got two solutions to reduce the generation loss, which are “SQ+CR” and “SQ+RL”. The quantitative evaluation results on different datasets and different baseline models are shown in Fig. 6. During the process of SIC, the image quality degradation quickly converges by using our improved mbt18 model, the average PSNR reduction is less than 1.0 dB on both datasets. After 50 times reencoding, we can still maintain excellent rate-distortion performance, even comparable to the first encoding of BPG. Experiments on cheng20 also demonstrate the effectiveness of our solutions.

Both solutions have their advantages and disadvantages. “SQ+CR” shows better performance in rate-distortion performance, but channel relaxation increases the model complexity at high bitrates. Without changing the original settings, “SQ+RL” can stop quality degradation faster during SIC, but the rate-distortion performance is slightly impacted due to the reversibility term, although it can be flexibly tuned by adjusting α. It is worth pointing out that the state-of-the-art multi-generation robustness level can be achieved using either CR or RL, they can be used simultaneously, i.e. “SQ+CR+RL”, but it will not show a more amazing result.

**VI. CONCLUSION**

This paper thoroughly analyzes the factors that affect the multi-generation robustness of LIC. We discovered and solved the quantization drift problem in existing models, and proposed two solutions to further reduce the generation loss. Extensive experiments show that our solutions are effective for different datasets and models. By using the model improved with our methods, images can maintain a considerable quality even after 50 times successive compression, while the rate-distortion performance is comparable to the first compression of BPG. The results are acceptable in practical applications, making the future of LIC more promising.

**ACKNOWLEDGMENT**

This work is financially supported by National Natural Science Foundation of China U21B2012 and 62072013, Shenzhen Cultivation of Excellent Scientific and Technological Innovation Talents RCJC20200714114435057, Shenzhen Science and Technology Program-Shenzhen Hong Kong joint funding project of SGDX2021112314440001. This work is also financially supported for Outstanding Talents Training Fund in Shenzhen.
Fig. 6. Quantitative evaluation results. Models based on cheng20 are evaluated with quality = 1, 3, 5.

REFERENCES

[1] G. K. Wallace, “The jpeg still picture compression standard,” IEEE Transactions on Consumer Electronics, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[2] F. Bellard, “Bpg image format,” https://bellard.org/bpg/, 2014, accessed: 2022-07-08.

[3] H. Fraunhofer, “Vvc official test model vtm,” https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/, 2019, accessed: 2022-07-08.

[4] 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 ICCV, 2021, pp. 14 677–14 686.

[5] Z. Guo, Z. Zhang, R. Feng, and Z. Chen, “Causal contextual prediction for learned image compression,” in IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 4, pp. 2329–2341, 2021.

[6] Y. Xie, K. L. Cheng, and Q. Chen, “Enhanced invertible encoding for learned image compression,” in ACM MM, 2021, pp. 162–170.

[7] F. Chen, Y. Xu, and L. Wang, “Two-Stage Octave Residual Network for End-to-End Image Compression,” AAAI, vol. 36, no. 4, pp. 3922–3929, Jun. 2022.

[8] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, “Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding,” in CVPR, 2022, pp. 5718–5727.

[9] W. Jiang, J. Yang, Y. Zhai, and R. Wang, “Multi-reference entropy model for learned image compression,” in ACM MM, 2021, pp. 339–353, 2020.

[10] Z. Cui, J. Wang, S. Gao, T. Guo, Y. Feng, and B. Bai, “Asymmetric gained deep image compression with continuous rate adaptation,” in CVPR, 2021, pp. 10532–10541.

[11] Z. Guo, Z. Zhang, and Z. Chen, “Deep scalable image compression via hierarchical feature decorrelation,” in 2019 Picture Coding Symposium (PCS). IEEE, 2019, pp. 1–5.

[12] L. Cao, A. Jiang, W. Li, H. Wu, and N. Ye, “Oodhdr-codec: Out-of-distribution generalization for hdr image compression,” AAAI, vol. 36, no. 1, pp. 158–166, Jan. 2022.

[13] X. Deng, W. Yang, R. Yang, M. Xu, E. Liu, Q. Feng, and R. Timofte, “Deep homography for efficient stereo image compression,” in CVPR, 2021, pp. 1492–1501.

[14] Y. Zhai, L. Tang, Y. Ma, R. Peng, and R. Wang, “Disparity-based stereo image compression with aligned cross-view priors,” in ACM MM, 2022, pp. 2351–2360.

[15] M. Li, K. Ma, J. Li, and D. Zhang, “Pseudocylindrical convolutions for learned omnidirectional image compression,” arXiv preprint arXiv:2112.11227, 2021.

[16] Z. Zheng, X. Wang, X. Lin, and S. Lv, “Get the best of the three worlds: Real-time neural image compression in a non-gpu environment,” in ACM MM, 2021, pp. 5400–5409.

[17] T. Richter, J. Keinert, A. Descampe, G. Rouvoy, and A. Willeme, “Multi-generation-robust coding with jpeg xs,” in IEEE International Symposium on Multimedia, 2017, pp. 6–13.

[18] J. Kim, S. Jang, J. Choi, and J. S. Lee, “Instability of successive deep image compression,” in ACM MM, 2020.

[19] D. Minnen, J. Ballé, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” NeurIPS, vol. 31, 2018.

[20] J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici, “Nonlinear transform coding,” IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 2, pp. 339–353, 2020.

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

[22] C. Horne, T. Naveen, A. Tabatabai, R. Eifrig, and A. Luthra, “Study of the characteristics of the mpeg2 4:2:2 profile-application of mpeg2 in studio environment,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 251–272, 1996.

[23] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Learned image compression with discretized gaussian mixture likelihoods and attention modules,” in CVPR, 2020, pp. 7939–7948.

[24] L. Theis, W. Shi, A. Cunningham, and F. Huszár, “Lossy image compression with compressive autoencoders,” in ICLR, 2018.

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

[26] E. K. Company, “Kodak lossless true color image suite,” http://r0k.us/graphics/kodak/, 1999.

[27] G. Toderici, W. Shi, R. Timofte, L. Theis, J. Ballé, E. Agustsson, N. Johnston, and P. Mentzer, “Workshop and challenge on learned image compression (clic2020),” CVPR, 2020. [Online]. Available: http://www.compression.cc