A Universal Encoder Rate Distortion Optimization Framework for Learned Compression

Jing Zhao[†][*], Bin Li[‡], Jiahao Li[‡], Ruiqin Xiong[†], Yan Lu[‡]
[†] Peking University, China
[‡] Microsoft Research Asia, China
{jzhaopku, rqxiong}@pku.edu.cn, {libin, jiahali, yanlu}@microsoft.com

Abstract

Learning-based image compression has drawn increasing attention in recent years. Despite impressive progress has been made, it still lacks a universal encoder optimization method to seek efficient representation for different images. In this paper, we develop a universal rate distortion optimization framework for learning-based compression, which adaptively optimizes latents and side information together for each image. The proposed framework is independent of network architecture and can be flexibly applied to existing and potential future compression networks. Experimental results demonstrate that we can achieve 6.6% *bit rate saving against the latest traditional codec, i.e., VVC, yielding the state-of-the -art compression ratio. Moreover, with the proposed optimization framework, we win the first place in CLIC validation phase for all the three different bit rates in terms of PSNR.*

1. Introduction

With the rapid development of deep learning, learning-based image compression has shown great potential and drawn increasing interests. Early works [5, 6, 7] utilize auto-encoders to compress images into latent representation, and estimate the rate as discrete entropy to make the network end-to-end trainable. After that, many powerful context-adaptive entropy models [20, 15, 11, 14, 13, 21] are developed to remove the redundancy in latent codes. In addition, some researchers explore efficient network structures, and introduce many modules, such as non-local residual block [16], attention block [19] and multi-scale fusion [11]. With these efforts, learning-based image compression exhibits a fast development in recent years. The state-of-the-art work [11] is even approaching the latest traditional compression standard, namely, VVC [22].

Although these networks have achieved impressive progress in image compression, they use a fixed trained encoder to compress all images, which cannot adapt to different image content. To address this issue, there are some works exploring image-specific rate-distortion optimization (RDO) to further improve learning-based compression. The work [18] proposes to adaptively update the encoder for different content. Furthermore, some works [17, 10] directly optimize the latent codes for each image and achieve more competitive performance. However, they simply exploit the existing optimizer designed for network training without considering the characteristics of latent codes.

To achieve higher compression ratio, we take the characteristics of compression process, especially the rounding effect, into consideration, and propose a new latent optimization strategy. In addition, inspired by the great potential of side information for traditional codecs [12, 24], we introduce side information at different levels, i.e., quantization step and post processing modulation scalar, into the learning-based compression, so as to further enhance the compression ratio. By integrating side information optimization with latent optimization, we develop a universal RDO framework for learning-based compression. The proposed optimization framework is independent of network architectures, and can be flexibly applied to the existing and future potential compression networks.

2. Approach

In this paper, we develop a universal encoder optimization framework for learning-based compression to adaptively boost the compression ratio for each image. The optimization framework consists of two main components, i.e., latent optimization and side information optimization.

2.1. Latent optimization

For the compression network, as soon as the training is completed, the encoder is determined, which would be applied to all the images to be compressed. However, due to

[*] This work was done when Jing Zhao was a full time intern with Microsoft Research Asia.

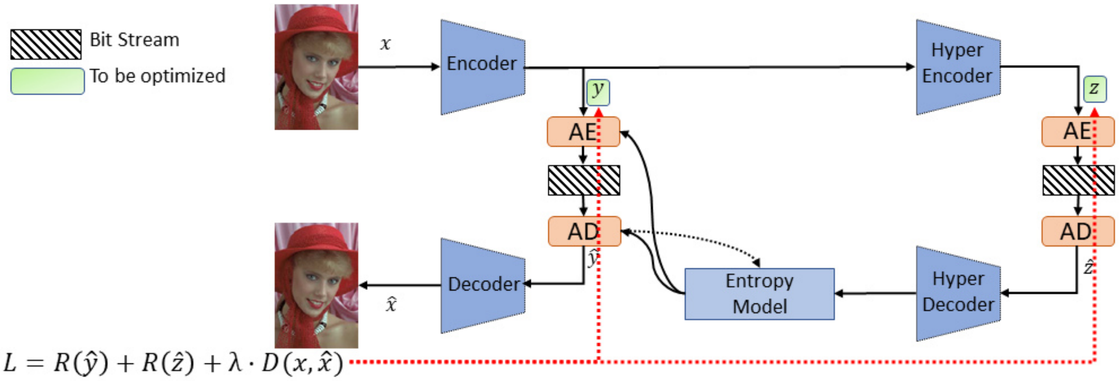

Figure 1. Illustration of latent optimization for learning-based compression. The network trained on a large dataset first encode the image x into a latent representation y and use hyperpriors z to model its dependency. Then a latent optimization is applied to further optimize y and z to adapt to its image content for higher compression ratio.

Algorithm 1 L-OPT

```
Input: Initial $y$ and $z$ from the network encoders
Output: Optimized $y_{\text{opt}}$ and $z_{\text{opt}}$
Initialize $\mathcal{L}_{\text{opt}} = 1e10$
for $t \leftarrow 1$ to $T$ do
    Estimate the loss $\mathcal{L}_t$ via Eq. (1)
    if $\mathcal{L}_t < \mathcal{L}_{\text{opt}}$ then
        $y_{\text{opt}} = y$, $z_{\text{opt}} = z$ and $\mathcal{L}_{\text{opt}} = \mathcal{L}_t$
    Update $y$ and $z$ via Eq. (2)
```

the diversity of image content, a fixed encoder cannot transform every image into its optimal latent representation, resulting in redundancy in latent codes.

To address this issue, we propose to further optimize the latent codes for each image. Fig. 1 provides a high-level overview of the proposed framework, which consists of two main components, i.e., a general compression network and latent optimization. Suppose x is the image to compress. It will be first encoded to a latent representation y with an encoder. Then, a hyper encoder is applied to capture the spatial dependency of y, producing hyper latent codes z. Sequentially, for higher compression ratio, the y and z are further optimized to adapt to its image content by minimizing the following loss function:

$$\mathcal{L} = \mathcal{R}(\hat{y}) + \mathcal{R}(\hat{z}) + \lambda \cdot \mathcal{D}(x, \hat{x}), \tag{1}$$

where $\hat{y}$ and $\hat{z}$ denote the quantized codes decompressed from bit stream. $\mathcal{R}(\hat{y})$ and $\mathcal{R}(\hat{z})$ represent the latent rate and hyper latent rate. $\hat{x}$ is the image reconstructed from $\hat{y}$ with the decoder. $\mathcal{D}(x, \hat{x})$ denotes the distortion, which can be measured with any differentiable evaluation metrics, such as PSNR and MS-SSIM.

In particular, different from the previous work [10] that simply use the Adam optimizer designed for training networks, we carefully analyze the characteristics of compression process, and derive an efficient latent optimization. Instead of adding uniform noise to approximate the quantization as the training, we use rounding to keep consistent with the practical encoding process and replace its derivative with a smooth approximation. To be specific, in the forward pass, we quantize the latent codes with rounding. In the backward pass, we use the gradient of identity function to replace the gradient of rounding to implement back propagation as introduced in [23]. Considering the effect of rounding, we should note the following issues. First, a small change of latent may not impact the behavior after rounding. For instance, 1.11 and 1.12 will be both rounded to 1. To guarantee the latents with large gradient can be properly updated, we adopt a fixed step to update the latent with the maximum gradient and update other latents proportionately. Second, for the latents with small gradients, a small change passing the rounding boundary may unexpectedly impact the behavior significantly. For example, 1.49 and 1.50 will be rounded to 1 and 2 respectively. To address this issue, in each iteration, we only update the latents with big gradients. To achieve high compression ratio, we jointly consider the two points and develop a content adaptive optimization mechanism, which is formulated as:

$$y := \begin{cases} y - \alpha \cdot \frac{y'}{|y'|_{\max}}, & |y'| > \beta \cdot |y'|_{\max} \\ y, & \text{otherwise} \end{cases} \tag{2}$$

Here y' denotes the gradient of y. α denotes the update step and β denotes the scalar determining update threshold, which are initialized to 0.8 and 0.25, respectively. They will be adjusted during the optimization. With the latent optimization, α decreases and β increases. We can also employ the same optimization mechanism to z, so as to optimize y and z together. The steps of the proposed latent optimization (L-OPT) is summarized in Algorithm 1.

Algorithm 2 QL-OPT

```
Input: Initial $Q_{\text{min}}$ and $Q_{\text{max}}$
Output: Optimized $q_{\text{opt}}$
Initialize $\mathcal{L}_{\text{opt}} = 1e10$
for $q \leftarrow Q_{\text{min}}$ to $Q_{\text{max}}$ by 0.05 do
    Optimize $y$ and $z$ with Algorithm 1.
    Calculate the loss $\mathcal{L}_t$
    if $\mathcal{L}_t < \mathcal{L}_{\text{opt}}$ then
        Update $q_{\text{opt}}$ and $\mathcal{L}_{\text{opt}}$
$Q_{\text{min}} = q_{\text{opt}} - 0.04$, $Q_{\text{max}} = q_{\text{opt}} + 0.04$
for $q \leftarrow Q_{\text{min}}$ to $Q_{\text{max}}$ by 0.01 do
    Optimize $y$ and $z$ with Algorithm 1
    Calculate the loss $\mathcal{L}_t$
    if $\mathcal{L} < \mathcal{L}_{\text{opt}}$ then
        Update $q_{\text{opt}}$ and $\mathcal{L}_{\text{opt}}$
```

2.2. Side information optimization

Considering traditional codecs use various side information to facilitate the decoding, we further introduce two different level side information, i.e., quantization step and post processing modulation scalar to our framework, and integrate side information optimization with latent optimization to enhance the compression ratio.

1) *Signal level side information.* The compression network usually sets the quantization step size to 1, and quantizes the latent codes by directly rounding them. However, 1 may not be the optimal quantization step for all the images. To address this issue, we introduce the quantization step q as the signal level side information to make the latent representation more efficient. Then, the quantized latents y_q and reconstructed latents $\hat{y}$ can be formulated as

$$\begin{aligned} y_q &= r(\frac{y-\mu}{q}) \\ \hat{y} &= y_q * q + \mu, \end{aligned} \tag{3}$$

where μ denotes the means predicted by entropy model and $r(\cdot)$ represents rounding function. Correspondingly, we should use $\mathcal{R}(y_q)$ to replace the $\mathcal{R}(\hat{y})$ in Eq. (1). To get the optimal q, we hierarchically search the candidate set. The joint quantization step and latent optimization (QL-OPT) algorithm is summarized in Algorithm 2.

2) *Post processing level side information.* Post processing networks are usually applied to further improve the reconstruction quality. However, considering the significant differences among the decoded images $\hat{x}$, it is not appropriate to use a fixed processing strength for all the images. To address this issue, we additionally introduce a post processing level side information, i.e., modulation scalar m to direct the post processing. Then the improved reconstruction $\tilde{x}$ can be formulated as

$$\tilde{x} = \mathcal{P}(\hat{x}, m), \tag{4}$$

where $\mathcal{P}(\cdot)$ denotes the post processing network. Here we use the pretrained DRUNet [26] as the post-processing network. We employ a hierarchical search, which is similar to Algorithm 2, to get the optimal modulation scalar m.

3. Experimental results

3.1. Implementation details

To facilitate the implementation of the proposed optimization, we develop a new compression network. The proposed network is modified from Cheng-Anchor[11] [1] and implemented based on CompressAI [8]. Different from the previous works that get both means and scales from the concatenated hyperpriors and context information, the entropy model of the proposed network infers means only based on hyperpriors and infers scales according to both hyperpriors and context information. Since the means are free from auto-regressive model, the proposed network can achieve parallel encoding, which is much faster than the sequential encoding in [11] and beneficial to the proposed rate distortion optimization. To train the proposed compression network, we use the Vimeo-90K dataset [25] and randomly cropped the images into 256×256 patches. The model was optimized using Adam optimizer with a batch size of 12. The learning rate is initialized to $1e-4$ and decreases during the training. The parameter λ of loss function belongs to the set $\{0.0018, 0.0035, 0.0067, 0.0130, 0.0250, 0.0483\}$.

3.2. Comparison with the state-of-the-art methods

To evaluate the compression methods, we compare our proposed compression framework with both traditional codecs, including JPEG2000 [3], HEVC [1] and VVC [4], and competitive learning-based compression methods, including Mean and Scale Hyperprior [20] and Cheng-Anchor [11]. For the learning-based compression methods, we use the released models from CompressAI [8] unless otherwise noted. We calculate the bit rate saving over the commonly used Kodak image set [2] to measure the compression ratio. In this paper, The bit rate saving is calculated by BD-rate as introduced in the work [9]. The rate is measured by bits per pixel (bpp), and the quality is measured by Peak Signal-to-Noise Ratio (PSNR).

Fig. 2 shows the RD curves of different compression methods. The released Cheng-Anchor is approaching the latest traditional codec, namely VVC. Our proposed network without optimization can achieve comparable results with the retrained Cheng-Anchor, which is slightly worse than the released Cheng-Anchor model due to lack of training details. It demonstrates that the modification of the entropy model do not degrade the compression ratio. However, as the means is free from the auto regressive model,

[1] Cheng-Anchor achieves the highest compression ratio among the pretrained models provided by CompressAI [8]

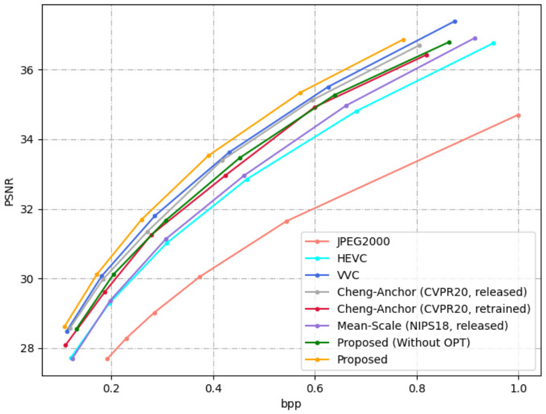

Figure 2. Comparison with the state-of-the-art methods on Kodak dataset. The proposed method achieves the best performance.

it is easier to speed up encoding and implement the proposed encoder optimization. The overall proposed framework, combining optimization with our learned compression network, outperforms all the state-of-the-art methods. Our framework achieves 6.6% bit rate saving against competitive VVC on Kodak dataset.

3.3. Applied to various compression networks

It is worth noting that proposed optimization algorithm is independent of network architectures. As long as the decoder is differentiable, it can be flexibly applied to existing and potential future compression networks as plugin. To demonstrate its flexibility and efficiency, we apply the proposed optimization to three different networks, i.e, Mean-Scale [20], Cheng-Anchor [11] and the proposed network (modified from Cheng-Anchor). As shown in Fig. 3, the proposed encoder optimization can significantly boost the compression ratio, which achieves 13.9%, 12.0% and 15.3% additional bit rate savings on the three different models, respectively.

3.4. Results on CLIC validation phase

We totally train 16 different compression models using the λ from 0.0008 to 0.0568 with a step ratio of about 4/3. The channel number is set to 128 and the size of each model is about 20M . Given a 250M decoder, we can save 11 compression models and a post posting model. In fact, for each targeting bit rate, we only use 3-5 compression models. The top results on the leaderboards are listed in Table 1. We group the results submitted from the teams with similar names (after removing the postfix such as PSNR and SSIM), and only list the results with the highest PSNR. With the proposed encoder RDO, We (DeepMC) can achieve the best PSNR performance for all the three targeting bit rates on CLIC validation dataset.

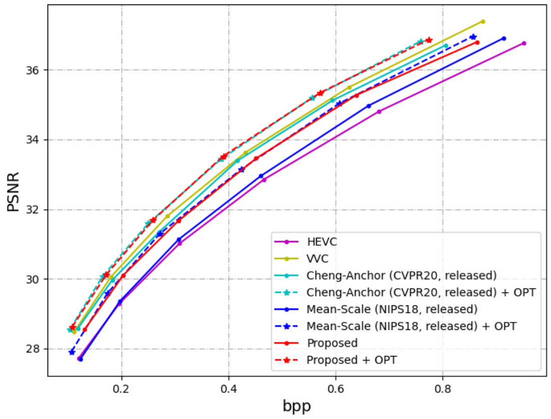

Figure 3. Applying the proposed optimization to various compression networks.

Table 1. The top PSNR results on CLIC leaderboards.

Team	0.075	0.150	0.300	Average
DeepMC	30.23	32.67	35.41	32.77
ANTxNN	30.14	32.47	35.31	32.64
VTM_VVC	29.26	31.58	34.70	31.85
IVPG	28.70	30.96	33.32	30.99
anf	28.63	30.80	33.32	30.92
HM_HEVC	28.31	30.50	33.90	30.90
SRCX_DLIC	27.84	30.83	34.01	30.89
wp2	28.26	30.37	33.17	30.60
IMCL_IMG	28.25	30.39	32.37	30.34
HHC	27.80	31.41	31.43	30.21

4. Conclusion

In this paper, we develop a universal encoder rate distortion optimization framework as plugin for learning-based image compression. Considering the diversity of image content, the proposed framework adaptively optimizes the side information and latents for each image to boost the learning-based compression ratio. Experimental results demonstrate that our proposed framework can remarkably boost the learning-based compression ratio, achieving more than 10% additional bit rate saving on three different network structures. The overall framework can achieve 6.6% bit rate saving against the latest VVC on Kodak dataset, yielding the state-of-the-art compression ratio. With the proposed optimization framework, we win the first place in CLIC validation phase for all the three targeting bit rates in terms of PSNR.

References

[1] HM, HEVC Reference Software. https://vcgit.hhi.fraunhofer.de/jvet/HM. Accessed: 2021-03-09. 3

[2] Kodak lossless true color image suite. http://r0k.us/graphics/kodak/. 3

[3] OpenJPEG, JPEG2000 Reference Software. https://jpeg.org/jpeg2000/software.html. 3

[4] VTM, VVC Reference Software. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM. Accessed: 2021-03-09. 3

[5] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281*, 2015. 1

[6] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimization of nonlinear transform codes for perceptual quality. In *2016 Picture Coding Symposium (PCS)*, pages 1–5. IEEE, 2016. 1

[7] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 1

[8] Jean Bégaint, Fabien Racapé, Simon Feltman, and Akshay Pushparaja. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020. 3

[9] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *VCEG-M33*, 2001. 3

[10] Joaquim Campos, Simon Meierhans, Abdelaziz Djelouah, and Christopher Schroers. Content adaptive optimization for neural image compression. *arXiv preprint arXiv:1906.01223*, 2019. 1, 2

[11] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7939–7948, 2020. 1, 3, 4

[12] Chih-Ming Fu, Elena Alshina, Alexander Alshin, Yu-Wen Huang, Ching-Yeh Chen, Chia-Yang Tsai, Chih-Wei Hsu, Shaw-Min Lei, Jeong-Hoon Park, and Woo-Jin Han. Sample adaptive offset in the hevc standard. *IEEE Transactions on Circuits and Systems for Video technology*, 22(12):1755–1764, 2012. 1

[13] Yueyu Hu, Wenhan Yang, and Jiaying Liu. Coarse-to-fine hyper-prior modeling for learned image compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11013–11020, 2020. 1

[14] Jan Klopp, Yu-Chiang Frank Wang, Shao-Yi Chien, and Liang-Gee Chen. Learning a code-space predictor by exploiting intra-image-dependencies. In *BMVC*, page 124, 2018. 1

[15] Jooyoung Lee, Seunghyun Cho, and Seung-Kwon Beack. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2018. 1

[16] Haojie Liu, Tong Chen, Peiyao Guo, Qiu Shen, Xun Cao, Yao Wang, and Zhan Ma. Non-local attention optimized deep image compression. *arXiv preprint arXiv:1904.09757*, 2019. 1

[17] Jerry Liu, Shenlong Wang, Wei-Chiu Ma, Meet Shah, Rui Hu, Pranaab Dhawan, and Raquel Urtasun. Conditional entropy coding for efficient video compression. *arXiv preprint arXiv:2008.09180*, 2020. 1

[18] 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*, pages 456–472. Springer, 2020. 1

[19] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4394–4402, 2018. 1

[20] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *NeurIPS*, 2018. 1, 3, 4

[21] David Minnen and Saurabh Singh. Channel-wise autoregressive entropy models for learned image compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3339–3343. IEEE, 2020. 1

[22] Jens-Rainer Ohm and Gary J Sullivan. Versatile video coding–towards the next generation of video compression. In *Picture Coding Symposium*, volume 2018, 2018. 1

[23] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017. 2

[24] Chia-Yang Tsai, Ching-Yeh Chen, Tomoo Yamakage, In Suk Chong, Yu-Wen Huang, Chih-Ming Fu, Takayuki Itoh, Takashi Watanabe, Takeshi Chujoh, Marta Karczewicz, et al. Adaptive loop filtering for video coding. *IEEE Journal of Selected Topics in Signal Processing*, 7(6):934–945, 2013. 1

[25] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019. 3

[26] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *arXiv preprint arXiv:2008.13751*, 2020. 3